

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

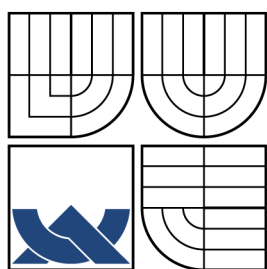
POPIS OBJEKTŮ V OBRAZE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

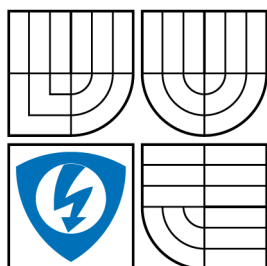
AUTOR PRÁCE
AUTHOR

BC. PAVEL DVOŘÁK

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ



FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

POPIS OBJEKTŮ V OBRAZE

OBJECT DESCRIPTION IN IMAGES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

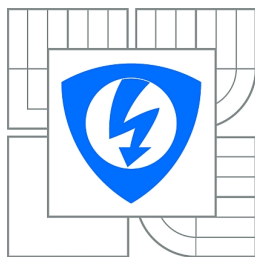
AUTOR PRÁCE
AUTHOR

BC. PAVEL DVOŘÁK

VEDOUCÍ PRÁCE
SUPERVISOR

ING. MARTIN ZUKAL

BRNO 2011



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Pavel Dvořák

ID: 70295

Ročník: 2

Akademický rok: 2010/2011

NÁZEV TÉMATU:

Popis objektů v obraze

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se se standardem MPEG-7 pro popis multimediálních dat. Zejména se zaměřte na deskriptory pro popis barev a tvaru objektu v obraze. Navrhněte a vytvořte vhodné datové struktury pro uchování těchto údajů v programu. Dále navrhněte a zrealizujte algoritmy pro výpočet jednotlivých vlastností. Získané údaje uložte do souboru v souladu se standardem MPEG-7.

DOPORUČENÁ LITERATURA:

[1] Bovik, A.I. (ed.). Handbook of Image and Video Processing. San Diego: Academic Press, 2000. ISBN 0121197905

[2] XIONG, Wei, et al Content-based medical image retrieval using dynamically optimized regional features. In Image Processing, 2005. ICIP 2005 : IEEE International Conference on. [s.l.] : [s.n.], 2005. s. 1232-1235. Dostupné z WWW: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1530621&isnumber=32662>>. ISBN 0-7803-9134-9.

[3] NASCIMENTO, J.; ABRANTES, A.; MARQUES, J. The Role of Middle Level Features for Robust Shape Tracking. In Proc. of 12th Portuguese Conf. on Pattern Recognition. Aveiro : [s.n.], June 2002. s. 339-352. Dostupné z WWW: <http://welcome.isr.ist.utl.pt/img/pdfs/367_recpad02b.pdf>.

Termín zadání: 7.2.2011

Termín odevzdání: 26.5.2011

Vedoucí práce: Ing. Martin Zukal

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato práce se zabývá popisem segmentů identifikovaných v obraze. Nejprve jsou zde popsány základní metody segmentace, protože se jedná o proces bezprostředně předcházející popisu objektů. Další kapitola je již věnována metodám, které se zaměřují na popis takto detekovaných regionů v obraze. Jsou zde prostudovány algoritmy používané pro charakterizaci různých vlastností objektů. Vyskytují se zde části věnované barvě, umístění, velikosti, orientaci, tvaru a topologii objektů. Konec této kapitoly je věnován momentům. Ve dalších kapitolách jsou navrženy vhodné algoritmy pro popis segmentů obrazu a vytváření XML souborů podle standardu MPEG-7 a následně popsána jejich implementace do prostředí RapidMiner. V poslední kapitole jsou zobrazeny a popsány výsledky implementace.

KLÍČOVÁ SLOVA

segmentace obrazů, popis objektu, zpracování obrazů, barva, umístění, velikost, tvar, topologie, obrazové momenty, měřítkové zakřivení, dominantní barva, MPEG-7

ABSTRACT

This thesis consider description of segments identified in image. At first there are described main methods of segmentation because it is a process contiguous before describing of objects. Next chapter is devoted to methods which focus on description identified regions. There are studied algorithms used for characterizing of different features. There are parts devoted to color, location, size, orientation, shape and topology. The end of this chapter is devoted to moments. Next chapters are focused on designing fit algorithms for segments description and XML files creating due to MPEG-7 standards and their implementation into RapidMiner. In the last chapter there are described results of the implementation.

KEYWORDS

image segmentation, image description, image processing, feature color, feature location, feature size, feature shape, feature topology, image moments, curvature scale space, CSS, dominant color, MPEG-7

DVOŘÁK P. *Popis objektů v obraze*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 62 s. Vedoucí diplomové práce Ing. Martin Zukal.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Popis objektů v obraze“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu diplomové práce Ing. Martinu Zukalovi za velmi užitečnou metodickou pomoc a cenné rady při zpracování projektu.

V Brně dne

.....

(podpis autora)

OBSAH

Úvod	10
1 Segmentace obrazů	12
1.1 Úvod	12
1.2 Metody segmentace	12
1.2.1 Prahování	12
1.2.2 Detekce hran	13
1.2.3 Regionální metody	13
1.2.4 Segmentace rozvodím	13
2 Standard MPEG-7	15
2.1 Úvod	15
2.2 Nástroje MPEG-7	15
3 Popis objektu	16
3.1 Úvod	16
3.2 Barva objektů	16
3.2.1 Dominantní barva	17
3.2.2 Rozvržení barev	17
3.3 Umístění objektu	17
3.4 Velikost objektu	18
3.5 Orientace objektu	20
3.6 Tvar objektu	20
3.6.1 Popis tvaru pomocí hraničních bodů	21
3.6.2 Popis tvaru pomocí regionu	25
3.7 Topologie objektu	29
3.7.1 Metody skeletonizace	29
3.8 Momenty	31
4 Návrh algoritmů	34
4.1 Popisované vlastnosti	34
4.1.1 Barva	34
4.1.2 Tvar	35
5 Implementace algoritmů	36
5.1 Prostředí RapidMiner	36
5.2 Dominantní barva	37
5.2.1 Postup výpočtu	38

5.3	Měřítkové zakřivení	39
5.3.1	Postup výpočtu	39
5.4	Excentricita	43
5.5	Kulatost	44
5.6	Zápis do XML dle MPEG-7	44
5.6.1	Zápis dominantní barvy	45
5.6.2	Zápis měřítkového zakřivení	47
6	Výsledky	49
6.1	Dominantní barva	49
6.2	Měřítkové zakřivení	50
6.3	Zápis do XML	52
7	Závěr	55
	Literatura	57
	Seznam symbolů, veličin a zkratk	60
	Seznam příloh	61
A	Obsah přiloženého CD	62
A.1	Návod na spuštění nástroje RapidMiner	62

SEZNAM OBRÁZKŮ

3.1	Řetězový kód takového objektu by byl, v případě, že se vychází z bodu označeného kolečkem, 0,0,0,1,0,0,3,3,2,3,2,2,1,2,2,1	22
3.2	Segmentace části hranice objektu při vyjádření pomocí polygonu . . .	24
3.3	Vytvoření CSS obrazu	25
3.4	Srovnání změny parametrů kulatost, konvexnost, plnost a kompaktnost v závislosti na tvaru objektu. Převzato z [20].	28
5.1	Ukázkové zapojení operátorů	37
5.2	Reprezentace <i>RGB</i> v paměti	38
5.3	Diagram výpočtu měřítkového zakřivení	40
5.4	Ukázka přepisu hraničních bodů do binárního obrazu a vytvoření hranice	41
5.5	Ukázka přepisu hraničních bodů do binárního obrazu a vytvoření hranice	42
6.1	Vizualizace výpočtu jedné dominantní barvy	49
6.2	Vizualizace výpočtu jedné dominantní barvy	49
6.3	Zobrazení výsledku výpočtu osmi dominantních barev pro jeden segment	50
6.4	Zobrazení vrcholů CSS obrazu	51
6.5	Vstupní obraz pro generaci XML souborů	52
6.6	Ukázka výstupního XML souboru s deskriptorem měřítkového zakřivení	53
6.7	Ukázka výstupního XML souboru s deskriptorem dominantní barvy .	54

ÚVOD

V posledních letech se velice rozmáhají digitální fotografie a obrazy a jejich zveřejňování na celosvětové síti Internet. Při vyhledávání obrázků pomocí různých vyhledávačů nebo i při vyhledávání ve vlastní domácí databázi je v současnosti kladen důraz na vlastnoručně vytvořené popisky. Někdy, zvláště v rámci Internetu, jsou do výsledků zahrnuty zcela irelevantní informace z důvodu špatného titulku, vytvořeného omylem, či úmyslně.

Tato práce se věnuje právě oblasti automatického popisu objektů obsažených v obrazech. Jedná se o část celkového procesu digitálního zpracování obrazů. Cílem celého projektu je automatické detekce objektů.

První část je věnována segmentaci obrazů, protože se jedná o proces bezprostředně předcházející popisu objektů. Nejprve je nutné obraz rozdělit na segmenty - objekty, se kterými se následně pracuje.

V další části práce jsou již rozebrány možnosti popisu objektů v závislosti na různých parametrech a to jak dle standardu MPEG-7, což je standard vytvořený skupinou MPEG (Moving Picture Experts Group), tak obecně. Těmito parametry mohou být: [20]

- barva,
- umístění,
- velikost,
- orientace,
- tvar.

Pro rozpoznání jednotlivých objektů je velmi důležitý jeho tvar. Z hlediska vnímání člověka je velmi důležitá i barva. Umístění, velikost a orientace jsou ovšem pro identifikaci rovněž potřebné.[23] Dalším parametrem, jenž souvisí s tvarem, je topologie. Jelikož přesný tvar je velmi složité popsat, je možné jej vyjádřit pomocí topologie, což práci usnadní.

Poslední část kapitoly o popisu objektů je věnována tzv. momentům. Jedná se o velice komplexní mechanismy, které v sobě mohou nést informace o všech zmíněných parametrech detekovaného regionu obrazu. [9]

Z množiny charakteristik představených v druhé kapitole budou v kapitole další vybrány ty, které se jeví vhodné pro popis obecných objektů. Jedná se o charakteristiky obsažené ve standardu MPEG-7.

Čtvrtá kapitola se již věnuje přímo implementaci algoritmů pro výpočet vybraných charakteristik do prostředí RapidMiner a vytvoření operátorů pro zápis výsledků do XML souborů.

V kapitole páté jsou popsány a zobrazeny výsledky aplikace těchto algoritmů na testovací obrazy a ukázány výstupní XML soubory.

1 SEGMENTACE OBRAZŮ

1.1 Úvod

Segmentace je velmi důležitým krokem při zpracovávání obrazu. Její výstupní data jsou daty vstupními pro krok zabývající se popisem objektů, který je hlavní náplní této práce. Proto je nutné se částečně seznámit i s touto problematikou.

Hlavním úkolem segmentace je rozčlenit obraz na několik částí se stejnými nebo podobnými vlastnostmi. V ideálním případě by toto rozdělení mělo být blízké tomu, jak by obraz rozdělil lidský mozek, tedy takové, že jednotlivé segmenty by přesně odpovídaly objektům obsažených v obraze. To však není v současné době reálně možné, je tedy snaha alespoň o přiblížení se tomuto případu.

Aspektem znesnadňujícím nejen segmentaci, ale i celkové digitální zpracování obrazu, je šum. Díky němu vznikají nepřesnosti v hraničním oddělení objektů. Existují algoritmy na jeho odstranění, ale ty také zvláště na rozhraní nebývají stoprocentní. Dalším problémem může být nehomogenní osvětlení a vzniklé stíny a rozdíly jasu.

Protože existuje několik metod segmentace, je nutné před jejím použitím zvolit, která bude použita. Děje se tak v závislosti na určité znalosti vstupních dat. Pro různé oblasti zpracování dat mohou být vhodné různé přístupy. Odlišné metody mohou pro různá vstupní data dosahovat rozdílných výsledků a ty, které se hodí například pro zpracování medicínských obrazů nemusí být vhodné pro zpracování obyčejných fotografií.

1.2 Metody segmentace

1.2.1 Prahování

Prahování patří ke starším a jednodušším metodám, která je však právě díky výpočetní nenáročnosti široce používána. Hlavním úkolem je zde vytvořit histogram jasu vstupních dat a následně nalézt práh, pro který platí, že pixely s větší hodnotou náleží popředí a pixely s hodnotou menší patří do pozadí.

Při hledání tohoto prahu je ovšem problém v tom, že histogramy reálných obrazů neobsahují ostré špičky a také jich obsahují víc, než jednu a je tedy nutné zvolit, který vrchol je ten správný.

Výstupem základního prahování je binární obraz, kde pixely s hodnotou 1 náleží popředí a ty s hodnotou 0 pozadí. Existují ale také různé modifikace této metody, které se liší také ve výstupních datech. [7]

1.2.2 Detekce hran

Hrana je oblast obrazu, kde dochází k významné změně intenzity jasu, či barvy. Nemusí ovšem nutně značit hranici mezi objekty ve scéně. Při detekci hran je na obrazová data nahlíženo jako na funkce. Metody se poté člení do dvou skupin podle toho, zda využívají její první nebo druhou derivaci.

Metody využívající první derivaci často využívají gradientních operátorů, jako Sobelův či Cannyho, a jsou tedy také nazývány gradientní metody. „Tyto operátory používají v jednom směru výpočet diferenciálu a ve směru kolmém používají filtry na potlačení vlivu šumu.“ [22] Výsledný hranový gradient je porovnán s prahem, který rozhodne, zda se jedná o hranu.

Není-li nutné znát směr a velikost hrany, je možné využít druhou skupinu metod založených na druhé derivaci. Druhá derivace funkce prochází při největší změně funkce nulou a právě tyto body jsou hledány. To je také snadnější než najít extrémy v případě derivace první. U této metody se často používá Laplaceův operátor. [22]

1.2.3 Regionální metody

Je-li v obraze přítomný šum, což je častý případ, je vhodnější využít techniky založené na zjišťování podobností pixelů. Podobnosti se mohou hledat v různých vlastnostech, např. jas, barva, textura.

Jednou z nejjednodušších metod této skupiny je rozšiřování oblasti. V tomto případě jsou nejprve rovnoměrně nebo náhodně rozmístěny inicializační pixely. Od nich se poté postupuje do okolí a segment vzniká rozrůstáním. Problémem této techniky je, že výsledky se mohou lišit v závislosti na počtu a rozmístění inicializačních pixelů.

Další regionální metodou je dělení a spojování oblastí. Obraz je rozdělen na čtyři kvadranty. Je-li kvadrant homogenní, je ponechán jako celek, v opačném případě je opět rozdělen na kvadranty a postup se opakuje. V případě, že některé sousedící čtverce tvoří homogenní oblast, jsou spojeny. [24]

1.2.4 Segmentace rozvodím

V případě segmentace rozvodím se na obrazovou funkci nahlíží jako na krajinu, kde jednotlivé hodnoty jasu označují nadmořskou výšku. Vodní hladina se zde postupně zvedá a jsou zaplavována údolí obrazové funkce. V místech, kde by se mohla údolí slít do jednoho, jsou vytvořeny hráze a body náležející této hranici jsou označeny vyšší hodnotou jasu. Pixely patřící do oblasti zaplavené vodou z jednoho pramene tvoří segment obrazu a jsou označeny stejným indexem. Algoritmus končí tehdy, jakmile jsou zaplaveny všechny oblasti. Nevýhodou této metody je, že produkuje

velké množství segmentů, proto se musí některé následně spojit, aby byly výsledky použitelné. [22]

2 STANDARD MPEG-7

2.1 Úvod

MPEG-7 je standard vytvořený skupinou MPEG (Moving Picture Experts Group). Tato skupina již dříve vyvinula standardy MPEG-1, MPEG-2 a MPEG-4. Od nich se však tento novější standard odlišuje. MPEG-7 je formálně známý jako „Multimedia Content Description Interface“ (Rozhraní pro popis multimediálního obsahu). Poskytuje mnoho standardizovaných nástrojů pro popis multimediálního obsahu, které jsou rozšiřitelné podle specifických potřeb aplikací. K přístupu (vyhledávání, filtrování, prohlížení) k multimediálnímu obsahu vytváří tato komplexní sada audiovizuálních popisových nástrojů vysoce účinný systém. [25][16]

Jsou zde rozlišeny a odděleny deskriptory pro popis vizuálního a zvukového obsahu. Vždy se jedná o textový popis, který je čitelný lidsky i strojově. To je dosaženo pomocí značkovacího jazyka XML. [25]

2.2 Nástroje MPEG-7

Sada nástrojů MPEG-7 se skládá z těchto částí: [16]

- deskriptory,
- popisová schémata,
- popisový jazyk,
- systémové nástroje.

Deskriptory reprezentují vlastnosti multimediálního obsahu. Definují syntaxi a sémantiku reprezentaci těchto vlastností.

Popisová schémata (Description schemes) specifikují strukturu a sémantiku vztahů mezi jednotlivými komponenty. Těmito komponentami mohou být jak deskriptory, tak právě popisová schémata.

Popisový jazyk (Description Definition Language - DDL) umožňuje vytváření nových popisových schémat a případně i deskriptorů a také rozšiřování a modifikaci již existujících popisových schémat.

Systémové nástroje jsou spojeny se synchronizací, transportem a ukládáním popisů.

Více informací jak o těchto nástrojích, tak o MPEG-7 všeobecně, může být nalezeno v [16] a [25].

3 POPIS OBJEKTU

3.1 Úvod

Popis objektů nalezených v obraze může být obecně rozdělen do čtyř tříd podle toho, jakou charakteristikou se zabývá. Jedná se o: barvu, umístění, velikost a tvar. Pro každou z těchto tříd je možné k měření přistupovat několika způsoby. Protože neexistuje obecně nejlepší přístup, vybrat ten správný je často problém. Pro odlišné typy objektů mohou být vhodné různé algoritmy vycházející navíc i z různých parametrů.

Při popisu umístění může být tedy například v určité situaci důležitá absolutní pozice v obraze, ale v jiném případě je naopak potřeba určit pozici relativně vzhledem k ostatním objektům. Stejně je to i s velikostí a barvou.

Protože popisování objektu je jen částí procesu zpracování obrazu, není jeho vstupem pouze původní obraz. Předchozím blokem procesu je segmentace. Je zde tedy nutné vědět, jak segmentace probíhala a jaká měla výstupní data. Jednotlivé regiony mohou být reprezentovány buď pomocí hraničních bodů nebo přiřazením každého pixelu do určité oblasti. I zde ovšem mohou také existovat určité rozdíly. Každý segment může být označen jiným číslem nebo barvou a všechny budou shrnuty v jednom obraze nebo může pro každou oblast existovat maska určující pouze danou část. [23] Na základě výše uvedeného se poté mohou jednotlivé algoritmy pro zpracování lišit.

V této kapitole budou představeny možnosti popisování nalezených objektů, nejen podle standardu MPEG-7, rozdělené podle charakteristiky, kterou se zabývají. Z popsanych metod budou v další kapitole vybrány ty, které se budou jevit jako vhodné pro popis obecných obrazů.

3.2 Barva objektů

Barva je jedna ze základních vlastností objektů využívaných k jeho popsání. Výhodou oproti ostatním vlastnostem popsaných níže je nezávislost na umístění v obraze, velikosti objektu a jeho posunutí a otočení a také na úhlu pohledu pozorovatele, tedy úhlu pořízení obrazu. Je ovšem závislá na umístění a typu zdroje světla ve scéně

Parametry vhodné pro popis barvy objektů mohou být následující:

- Dominantní barva
- Rozvržení barev

3.2.1 Dominantní barva

Dominantní barva podává kompaktní informace o barvě objektu. Definována je dle vztahu 3.1, kde N označuje počet dominantních barev, c_i vektor barvy (závisí na zvoleném barevném prostoru), p_i procento výskytu barvy, v_i proměnlivost hodnot v odpovídající barvě (pro každou barvu barevného prostoru jedna hodnota), a s_i celkovou prostorovou homogenitu dominantní barvy. Parametry v_i a s_i jsou volitelné. [27]

$$F = \{(c_i, p_i, v_i), s_i\}, \text{ kde } i = 1, 2, \dots, N \quad (3.1)$$

3.2.2 Rozvržení barev

Nalezení rozvržení barev v objektu se skládá z více částí. Nejprve je nutno region rozdělit na bloky. Jejich velikost je závislá na požadovaném rozlišení. Následně se v těchto blocích vybere reprezentativní barva. Toho lze dosáhnout několika metodami, snadné je například vybrat barvu průměrnou nebo medián hodnot. Pokud blok přesahuje hranici objektu, barevné hodnoty pixelů nacházejících se mimo objekt nejsou započítány. [27]

3.3 Umístění objektu

Pro určení polohy objektu existuje několik různých definic. Některé jsou spočitatelné jednodušeji, jiné obtížněji. Za základní rozdělení definic může být považováno určení jedním bodem, kde se určí "střed" objektu, nebo určení omezením, které spočívá v jednoduchém vymezení oblasti výskytu v obraze. Dále je také rozdíl v tom, zda je uvažována absolutní nebo relativní pozice vzhledem k dalším segmentům.

Jednoduchým způsobem vytyčení objektu je být nalezení maximální a minimální polohy v horizontálním a vertikálním směru. Takto se snadno určí buď hraniční obdélník objektu a lehce se z těchto hodnot vypočítá jeho střed, který může také sloužit k reprezentaci pozice. Pro komplexnější vyjádření polohy ovšem většinou určení pomocí středu nestačí. Výsledek může být totiž snadno zkreslený několika málo pixely. Oba dva takto získané parametry (střed a hraniční obdélník) jsou velice citlivé na orientaci objektu. K předejití této závislosti je možné definovat opsanou kružnici, kdy středem této kružnice je geometrický střed objektu.

Posledním bodem, určujícím polohu, může být tzv. centroid. Ten může být vypočítán z hraničních bodů nebo za použití všech pixelů uvnitř objektu. Umístění

tohoto bodu je vypočítáno jako aritmetický průměr souřadnic všech pixelů v každé ose podle rovnice 3.2. [20]

$$C[x, y] = \left[\frac{\sum x_i}{plocha}, \frac{\sum y_i}{plocha} \right], \quad (3.2)$$

kde *plocha* znamená celkový počet pixelů. Jelikož se v rovnici dělí, může být centroid spočítán s přesností lepší než pixelovou, což je někdy vhodné. Pokud je použita reprezentace objektu pomocí hraničního vyjádření, vypočítá se pozice tohoto bodu ze složitější rovnice 3.3. [20]

$$C[x, y] = \left[\frac{\sum (x_i + x_{i-1})^2 \cdot (y_i + y_{i-1})}{plocha}, \frac{\sum (y_i + y_{i-1})^2 \cdot (x_i + x_{i-1})}{plocha} \right] \quad (3.3)$$

kde x_i, x_{i-1} , resp. y_i, y_{i-1} jsou souřadnice sousedních hraničních bodů. I *plocha* se zde ovšem počítá jiným způsobem, a to podle rovnice 3.4. [20]

$$Plocha = \frac{\sum (x_i + x_{i-1}) \cdot (y_i + y_{i-1})}{2} \quad (3.4)$$

To, jak moc se od sebe liší geometrický střed a centroid, závisí na tvaru segmentu. U objektů pravidelného tvaru, např. kruhu, čtverce, elipsy, budou mít tyto dva body shodné umístění. U nepravidelných tvarů bývá často umístění velmi rozdílné. Příklad i podrobnější informace o výpočtech umístění mohou být nalezeny v [20].

Sousedství

Do vyjádření pozice segmentu je zahrnuto také relativní umístění vzhledem k ostatním částem obrazu. Zde je zajímavé například spočítání objektů, se kterými ten zkoumaný přímo sousedí. Jsou-li části odlišeny identifikačním číslem, je možné toho dosáhnout pomocí tabulky sousedících částí. Ta je vytvořena tak, že pro každý hraniční bod objektu se zkoumají sousední pixely. Má-li nějaký z jeho sousedů takové identifikační číslo, pro které ještě v tabulce sousedů není záznam, je toto sousedství vloženo. Po projití všech hraničních bodů, je již třeba jen spočítat záznamy v tabulce. [23]

3.4 Velikost objektu

Jednoduchým vyjádřením velikosti zkoumaného objektu v obraze je spočítání příslušných pixelů. Je-li segment vyjádřen pomocí hraničních bodů, je možné jeho velikost spočítat podle již zmíněné rovnice 3.4.

Ne vždy je ovšem třeba vypočítat pouze počet bodů náležejících segmentu, ale například pokud se uvnitř objeví nějaká prázdná místa, je potřeba je také zahrnout do oblasti pokryté touto částí obrazu a tudíž spočítat celkovou rozlohu, kterou

zkoumaný objekt pokrývá. Taková situace může nastat například, je-li na obraze dům a je nutné spočítat jeho velikost. Okna, která s velkou pravděpodobností nebudou součástí segmentu vyjadřující tuto budovu, jsou ale jeho součástí a do velikosti domu je třeba je též zahrnout. Rozhodnutí o tom, zda se „díry“ uvnitř objektu mají zahrnout do jeho rozlohy nebo ne, je obtížné a nechává se již na člověku, který zná další souvislosti obrazu.

Třetí možností vyjádření velikosti je rozloha nejmenšího konvexního mnohoúhelníku, který obklopuje patřičný segment. Takovýto mnohoúhelník je možné vytvořit pomocí rotace. S objektem je postupně po stanovených krocích otočeno o 180° . V každém kroku jsou spočítány nové souřadnice všech bodů a hledána minima a maxima ve směru osy x . Po ukončení procesu rotace označují tyto mezní body vrcholy konvexního mnohoúhelníku. Úhel rotace a tedy i počet vrcholů se volí v závislosti na přesnosti, které je třeba dosáhnout. Jelikož je poté objekt reprezentován pomocí hraničních bodů, jeho velikost se vypočítá opět podle již zmiňované rovnice 3.4.

Některé algoritmy používají pro popis a následné porovnávání velikosti průměr kruhu o obsahu odpovídajícímu rozloze objektu. Tato hodnota se jednoduše spočítá ze základního vzorce pro výpočet obsahu kruhu. Mnohem lepším přístupem ovšem je nahradit kruh elipsou, čímž se docílí nejen vyjádření velikosti, ale také aproximace tvaru, orientace atd. [20]

Obvod

Pro vyjádření velikosti je možné kromě obsahu plochy využít i obvod. Je-li segment vyjádřen pomocí všech náležejících pixelů, je třeba nejprve zjistit hraniční body, které sousedí s pozadím, je-li popsán hraničními body, je tento krok přeskočen. Přístupovat k problému popisu obvodu je možné dvěma způsoby.

První z nich pouze počítá krajní body, přesněji počet hran, protože pixel nacházející se v rohu, je zahrnut dvakrát. Pro druhou metodu je vhodné vyjádření pomocí řetězového kódu, který je vysvětlen v kapitole zabývající se tvarem. Hlavní rozdíl je v tom, že při výpočtu délky šikmé hrany není délka rovna dvěma jednotkám, ale pouze $\sqrt{2}$, což vnáší do výsledku větší přesnost. Vzorec, který je využit k vypočítání obvodu je uveden v rovnici 3.5. [20]

$$O = \sum \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \quad (3.5)$$

Při pohledu na rovnici je jasné, že se vlastně sčítají vzdálenosti mezi sousedními hraničními body. Jednodušší určení je v případě reprezentace obrazu pomocí řetězového kódu, který je popsán později. Zde je každé hodnotě kódu přiřazena vzdálenost buď 1 nebo $\sqrt{2}$ a do výsledného obvodu je stačí již pouze sečíst. Stejně jako v případě určování velikosti, je zde nutné rozhodnout, zda se bude počítat i obvod vnitřních

částí nenáležících do segmentu nebo ne a zda se bude určovat přesná délka hranice nebo obvod nejmenšího konvexního mnohoúhelníku.

Problémem, na rozdíl od předchozích vlastností, je závislost tohoto parametru na rozlišení obrazu. Je-li rozlišení zmenšováno, klesá také počet nepravidelností ve tvaru a tím pádem se i celkový obvod. Další potíže jsou způsobeny v případě přítomnosti šumu, který vnáší nerovnosti do okraje objektu a s tím roste délka hranice.

Kvůli výše zmíněným problémům není tento parametr považován za příliš vypovídající při popisu, někdy je ale vhodné jej vypočítat.[20]

3.5 Orientace objektu

Orientaci objektu je vhodné vyjádřit jen pro objekty, které jsou roztažené, a může být, stejně jako všechny ostatní charakteristiky, počítána různě. Jedním z přístupů je nahrazení objektu elipsou. Hlavní osa elipsy vyjadřuje největší vzdálenost libovolných dvou bodů segmentu pomocí otáčení. Je možné ji určit stejně, jako v případě hledání konvexního mnohoúhelníku. Je zde ale vybrána pouze jediná dvojice bodů. Z jejich vzdálenosti a celkové velikosti objektu je podle matematického vzorce pro výpočet obsahu elipsy odvozena vedlejší osa. Pro zjištění orientace je pouze určen úhel otočení v daném kroku, při němž byly tyto dva body získány. I zde ovšem mohou nastat problémy s přesností. Pro představu může být uveden objekt tvaru obdélníku. Je-li vypočítána jeho orientace pomocí elipsy, bude mít sklon podle jedné z úhlopříček, což ovšem neodpovídá skutečnosti.[20]

Lepším postupem je vypočítání orientace pomocí momentů, jenž jsou uvedeny později, nebo pomocí metody používané pro výpočet roztažení objektu, jejímž vedlejším výsledkem je právě úhel natočení objektu. Toto téma je zmíněno zde 3.6.2.

3.6 Tvar objektu

Popis tvaru je obecně nejobtížnější úlohou v části popisu objektu při zpracovávání obrazu. Všechny předchozí vlastnosti je možné vyjádřit numericky jedním nebo několika čísly, zde to ovšem neplatí. I pro člověka samotného se jedná o nesnadný úkol. Nejprve je nutné určit, které parametry jsou v dané situaci důležité. To se může lišit obraz od obrazu, či objekt od objektu.

Možná právě proto, že není snadné určit tvar objektu, je tento parametr nejdůležitějším ze všech. Pokud je změněna velikost či barva jakékoliv věci nebo je o nějaký úhel otočena, stejně je pro lidské oko snadné určit o co se jedná. Pro představu je možné uvést příklad: Mějme na obrázku klokana. Nabarvíme-li ho na fialovo a desetkrát ho zmenšíme oproti okolí, stejně dokážeme určit, o jaké zvíře se

jedná. Provedeme-li to ovšem obráceně, tedy změníme-li pouze jeho tvar, tak podle velikosti a barvy nemůžeme klokana určit.

Člověk často při popisování tvaru užívá přirovnání. Každý určitě někdy při pohledu na oblohu uviděl a přirovnal mraky k nějaké věci, zvířeti či něčemu jinému. Děje se tak, díky zkušenostem lidí, díky tomu, že již někdy určitou věc viděli a zapamatovali si její tvar. Jsou pak schopni objekty podle tvaru rozeznávat. Pokud by obrázek s klokánem dostalo do ruky dítě, které o něm již sice slyšelo, ale ještě nikdy jej nevidělo ani na fotce nebo vidělo, ale nikdo mu neřekl o co se jedná, nedokázalo by jej poznat.

Stejně tak i u počítačového vidění se nejčastěji pro rozpoznávání objektů používá jejich tvar a to tak, že jsou vytvořeny určité šablony pro každý rozpoznávaný objekt, a následně je zkoumáno, k jakému tvaru v paměti se zkoumaný objekt nejvíce blíží. Tyto metody jsou ale nad rámec této práce, proto zde nebudou popisovány.

Velkým problémem pro rozpoznávání objektů v obraze podle tvaru je orientace. U většiny předmětů z reálného života je důležitý úhel, pod kterým je daná věc pozorována. S úhlem se totiž mění i tvar objektu, což znesnadňuje identifikaci.

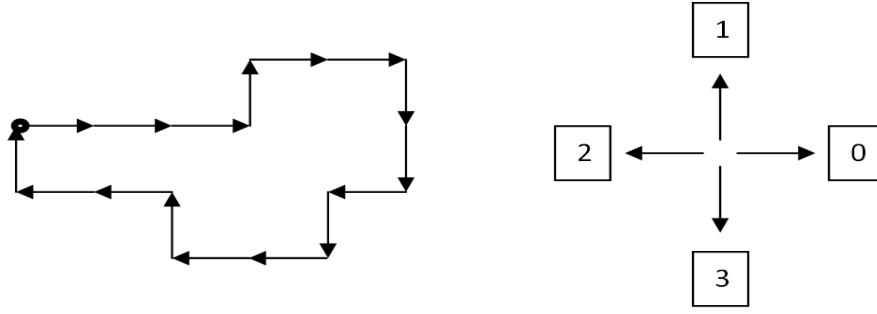
Další potíže nastávají, je-li na obraze vidět pouze část zkoumaného objektu, je-li buď překryt jiným nebo je umístěn v okrajových částech tak, že jeho další část se nachází mimo zobrazenou plochu. V tomto případě se využívá tzv. lokálních deskriptorů, které zkoumají pouze části regionů. Naproti tomu existují globální deskriptory, které jsou určené pro popis celého objektu, ty v tomto případě ale nemohou být použity. Proto je tedy důležité nejdříve rozhodnout, které z těchto deskriptorů se mají využít. [20] [23]

3.6.1 Popis tvaru pomocí hraničních bodů

Řetězový kód

Řetězový kód, známý také jako Freemanův kód, využívá pro popis tvaru řetězec číselných hodnot. Ty jsou odvozeny z hraničních bodů. Pro každý hraniční pixel existuje nějaký soused. Hodnota, která je pro tento bod uložena značí, kterým směrem se jeho soused nachází. V závislosti na vstupních datech, zda jsou uvažováni čtyři nebo osm sousedů, je možný počet různých hodnot čtyři, respektive osm. Princip je vysvětlen na obrázku 3.1.

Takovýto řetězový kód je závislý na otočení objektu, pro vyhnutí se této závislosti je možné použít rozdílový řetězový kód, kde se jednotlivé hodnoty vyjadřují jako rozdíl mezi aktuálním a předchozím. [23]



Obr. 3.1: Řetězový kód takovéhoto objektu by byl, v případě, že se vychází z bodu označeného kolečkem, 0,0,0,1,0,0,3,3,2,3,2,2,1,2,2,1

Zakřivení

Parametr zakřivení je definován jako poměr mezi počtem změn ve sklonu hranice a její délkou a je také nazýván jako rovnost hranice. Jedná se o koeficient mezi počtem všech hraničních bodů a hraničních bodů takových, kde se výrazně změnil směr hranice. Čím menší toto číslo je, tím je hranice rovnější.

Snadno může být tento parametr vypočten z řetězového kódu, kde změna ve směru sklonu je indikována změnou v kódu. Změny o různé úhly, v závislosti na rozdílu sousedních hodnot v řetězovém kódu, mohou být také vyneseny do histogramu, který má také vypovídající hodnotu o zakřivení hranice objektu. [23]

Tětivové rozdělení

Velmi dobrou metodou pro popis tvaru může být tětivové rozdělení. Jedná se o histogram vzdáleností a úhlů mezi všemi hraničními body, tedy délek a úhlů všech tětiv. Zatímco délky tětiv jsou závislé na velikosti objektu a nezávislé na jeho otočení, s úhly je to přesně naopak. Rozdělení délek tětiv se vypočítá podle rovnice 3.6. [23]

$$h(\Delta x, \Delta y) = \sum_i \sum_j b(i, j) b(i + \Delta x, j + \Delta y), \quad (3.6)$$

kde $b(i, j) = 1$ pro všechny hraniční body, pro všechny ostatní má hodnotu nulovou. V rovnici 3.7 je uveden vzorec pro výpočet úhlové rozdělení.

$$h_\alpha(\Theta) = \int_0^{\max(r)} h(\Delta x, \Delta y) dr, \quad (3.7)$$

kde r udává vzdálenost mezi danými body. Podrobnější popis je možné najít v [23].

Polygony

Další algoritmus pro vyjádření tvaru využívá k popisu polygony, které aproximují hranice objektů. Jedná se tedy vlastně o rozdělení hranice na několik segmentů. Hlavním úkolem je zde vypočítání vrcholů mnohoúhelníku, jejichž počet závisí na požadované přesnosti aproximace. Pro získání vrcholů existuje několik různých přístupů.

Jedním z nich je postupný průchod okrajem regionu. Na začátku je zvolen výchozí bod. Následně se prochází hraničními body, dokud je dodržena podmínka rovnosti, jsou body přidávány do dané části hranice. Jakmile je tato podmínka porušena, bod, který ji porušil, je označen jako vrchol polygonu a předchozí krok se opakuje. Existuje několik variant této metody.

Dalším přístupem je získávat vrcholy pomocí tzv. tolerančního rozpětí. Podle požadované přesnosti se zde nastaví velikost rozpětí, čím menší, tím větší přesnost. Průchod hraničními body je podobný předchozímu. Je zde ale určeno toleranční pásmo. Pokud se okrajové pixely nacházejí uvnitř něj, náleží stále jednomu segmentu hranice, bod, který se dostane mimo, je opět označen jako nový vrchol. Problémem zde ovšem je, že vrchol mnohoúhelníku by měl být zvolen již o několik kroků dříve. Proto se zde využívá vylepšení algoritmu pomocí dělení částí hranic. Může zde být tedy vytvořené větší toleranční pásmo a později se pomocí rozdělení segmentů hranice přesnost zvýší.

Při segmentaci se toleranční pásmo zmenší. Následně jsou mezi jednotlivými vrcholy mnohoúhelníku vytvořeny úsečky a počítány vzdálenosti od náležejících hraničních pixelů. Leží-li všechny v požadovaném tolerančním pásmu, jedná se o konečné vyjádření této části hranice objektu. Pokud tomu tak není, je nalezen hraniční bod s největší vzdáleností od úsečky a zvolen jako nový vrchol. Tím je část mezi původními dvěma vrcholy změněna na dvě nové a dělení je prováděno znovu.[23]

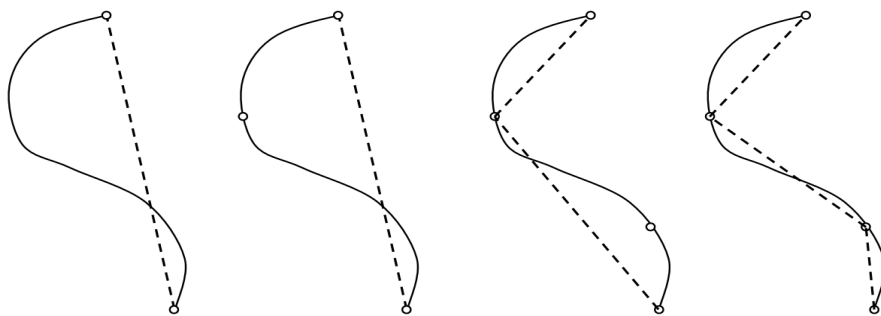
Postup je naznačen na obrázku 3.2 .

Neměnné křivky

Další možností jak rozdělit hranici na několik částí je pomocí neměnných křivek, reprezentovaných polynomy, většinou druhého řádu, jako kružnice, parabola atd. Každý segment hranice je poté vyjádřen pomocí jednoho z těchto polynomů. [23]

Měřítkové zakřivení

Měřítkové zakřivení (Curvature Scale Space - CSS) je výkonný algoritmus pro popis a detekci objektů. Principem této metody je rozdělení hraniční křivky na konvexní a konkávní oblasti a nalezení inflexních bodů, tj. bodů, kde je křivost hranice nulová,



Obr. 3.2: Segmentace části hranice objektu při vyjádření pomocí polygonu

tedy konvexní část přechází do konkávní a obráceně. To je prováděno v několika cyklech, kdy je hranice postupně vyhlazována a tyto inflexní body postupně mizí. Vyhlazování hranice je dosaženo opakovaným filtrováním jednorozměrným Gaussovým filtrem pro horizontální i vertikální souřadnice. [16] [17]

Pro výpočet křivosti je využito prvních a druhých derivací obou souřadnic obrazu podle pozice na hraniční křivce. Vychází se z rovnice 3.8

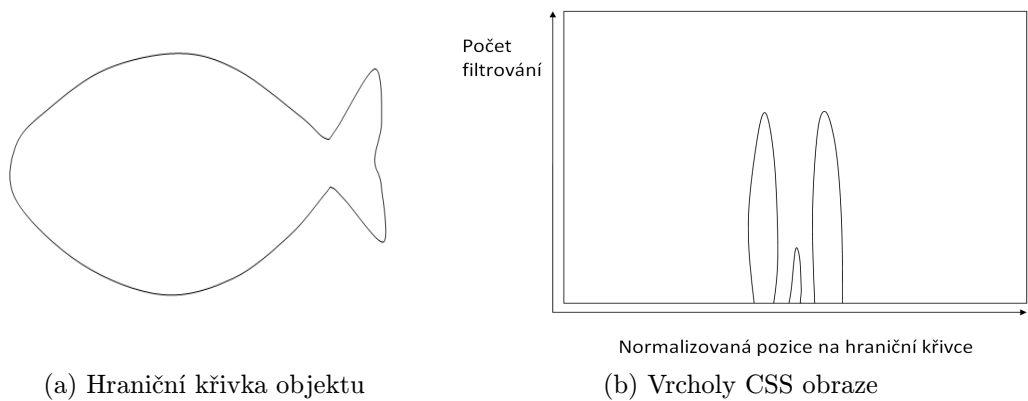
$$\kappa(s) = \frac{(x'(s)y''(s) - x''(s)y'(s))}{(x'(s)^2 + y'(s)^2)^{3/2}}, \quad (3.8)$$

kde κ je křivost v daném bodě, s popisuje pozici bodu na hraniční křivce, x a y vyjadřují osy obrazu.

Pro nalezení inflexních bodů ovšem není třeba přesná hodnota, ale pouze znaménko výsledku tohoto výpočtu. Inflexní bod je následně nalezen v místě změny znaménka. [16] Vyhledávání inflexních bodů je prováděno po každé aplikaci gaussovského filtru. Jakmile již není žádný takový bod nalezen, tedy objekt je plně konvexní, je proces zastaven.

Tímto postupem je vytvořen tzv. CSS obraz (CSS image), jehož horizontální souřadnice určuje normovanou pozici na hraniční křivce a vertikální souřadnice vyjadřuje počet aplikovaných filtrů. Příklad vytvoření CSS obrazu je uveden na obrázku 3.3. [16] U tohoto příkladu je za počáteční pozici zvolen bod nejvíce vlevo a hranice je procházena ve směru hodinových ručiček. U originálního obrazu je možné si povšimnout tří konkávních oblastí, které ve výsledku vytváří v CSS obraze tři vrcholy. Tyto vrcholy jsou pro popis objektu stěžejní.

Každý inflexní bod je následně vyjádřen pomocí vektoru, jehož horizontální souřadnice udává relativní vzdálenost od zvoleného počátečního bodu na hraniční křivce vzhledem k délce hranice a vertikální souřadnice vyjadřuje normalizovaný počet vyhlazování nutných k odstranění tohoto bodu vzhledem k maximu těchto hodnot. [16]



Obr. 3.3: Vytvoření CSS obrazu

Algoritmus je poměrně odolný vůči šumu, protože šumem nezmizí konkávní oblasti, mohou být pouze mírně posunuté. Dále je také invariantní vůči změně měřítka a také vůči rotaci objektu. Nezávislost na měřítku je dosažena normalizací obou výstupních hodnot. Invariantnost vůči rotaci je dosažena tím, že inflexní body si mezi sebou udržují pořád stejnou vzdálenost.

3.6.2 Popis tvaru pomocí regionu

K popisu tvaru je také možné použít region samotný. Na rozdíl od předchozích není možné v tomto případě opět podle popisu objekt zrekonstruovat, což ale v tomto případě není důležité. Není zde popisován tvar jako takový, ale jeho vlastnosti. V následující části jsou popsány vlastnosti objektů, které mohou být užitečné pro další zpracování obrazu a rozpoznání objektů v něm obsažených. Bližší informace o všech následujících parametrech je možné najít v literaturách [20] [23]. V závislosti na literatuře se názvy i definice jednotlivých parametrů mohou lišit.

Činitel tvaru

Činitel tvaru popisuje nepravidelnost tvaru objektu. Rozsah hodnot tohoto parametru je $t \in (0; 1)$, čím vyšší je hodnota, tím méně je nepravidelností. Je možné jej vypočítat podle rovnice 3.9. [20]

$$t = \frac{4\pi \cdot S}{o^2} \quad (3.9)$$

Parametry obvod o a plocha S jsou vysvětleny v kapitole 3.6.

Kulatost

Parametr kulatost značí, jak moc je tvar objektu podobný kruhu. Hodnoty se pohybují ve stejném rozmezí jako u činitele tvaru. Vyšší hodnota znamená větší přiblížení k tvaru kruhu. Vzorec pro výpočet je uveden v rovnici 3.10. [20]

$$c = \frac{4 \cdot S}{\pi \cdot p^2}, \quad (3.10)$$

kde c značí kulatost, S plochu regionu a p jeho maximální průměr.

Roztažení

Vlastnost roztažení vyjadřuje podíl mezi délkou a šířkou ohraničujícího obdélníku, což je podle standardních metod obdélník s nejmenším obsahem, který ohraničuje daný objekt. V tomto roce však byla navržena nová metoda, zabývající se nalezením tohoto obdélníku a výpočtem parametru roztažení, která se od předchozích liší v tom, že ohraničující obdélník není ten s nejmenším obsahem, ale ten s největším roztažením. Bližší informace mohou být nalezeny v [8].

Tato metoda vychází z toho, že ohraničující obdélník $R(S)$, kde S vyjadřuje konvexní obálku regionu, obsahuje vždy minimálně jednu hranu z této konvexní obálky. To ulehčuje jeho nalezení. K výpočtu roztažení je následně použita rovnice 3.11

$$r = \frac{d}{s} \quad (3.11)$$

kde r je vypočtený parametr roztažení, délka d a šířka s vyjadřují delší, resp. kratší, stranu ohraničujícího obdélníku.

Konvexnost

Konvexnost popisuje podíl mezi obvodem objektu a obvodem nejmenšího konvexního mnohoúhelníku, který ohraničuje objekt. Jeho vytvoření již bylo uvedeno v předchozí kapitole. Výpočet je prováděn podle rovnice 3.12. [20]

$$k = \frac{o_o}{o}, \quad (3.12)$$

kde k je konvexnost, o_o označuje obvod konvexní obálky segmentu a o vyjadřuje obvod segmentu.

Plnost

Plnost je parametr podobný konvexnosti. Jeho výpočet se ovšem provádí z odlišných vlastností regionu, jak je vidět v rovnici 3.13

$$f = \frac{o}{S_o}, \quad (3.13)$$

kde f je parametr plnost, o značí obsah segmentu a S_o obsah obálky, proto se hodnoty mohou s různými tvary měnit odlišně. Srovnání těchto parametrů je uvedeno na obrázku 3.4. [20]

Kompaktnost

Kompaktnost je opět parametr podobný předchozím avšak s odlišným výpočtem. V rovnici 3.14 je uveden způsob jeho výpočtu a na obrázku 3.4 je opět srovnána jeho proměna v závislosti na tvaru vstupního objektu. [20]

$$k = \frac{\sqrt{\frac{4}{\pi} \cdot S}}{d} \quad (3.14)$$

V této rovnici d znovu označuje maximální průměr a S rozlohu regionu, zatímco k nyní vyjadřuje počítaný parametr kompaktnost.

Rozlehlost

Parametr rozlehlost vyjadřuje podíl mezi čistou rozlohou objektu bez vnitřních „děr“ a rozlohou nejmenšího ohraničujícího obdélníku, výpočet je uveden v rovnici 3.15. [20]

$$r = \frac{S}{S_o}, \quad (3.15)$$





kde r je rozlehlost, S vyjadřuje obsah segmentu, S_o obsah ohraničujícího obdélníku.

Excentricita

Dalším parametrem popisujícím tvar objektu na základě jeho vnitřních bodů je excentricita neboli výstřednost. Výpočet je založen na vzdálenosti jednotlivých bodů od centroidu regionu a počítá se dle rovnice 3.16

$$e = \frac{\sqrt{i_{20} + i_{02} + \sqrt{i_{20}^2 + i_{02}^2 - 2i_{20}i_{02} + 4i_{11}^2}}}{\sqrt{i_{20} + i_{02} - \sqrt{i_{20}^2 + i_{02}^2 - 2i_{20}i_{02} + 4i_{11}^2}}}, \quad (3.16)$$

kde e je počítaný parametr excentricita, $i_{02} = \sum_{k=1}^M (y_k - y_c)^2$, $i_{11} = \sum_{k=1}^M (y_k - y_c)(x_k - x_c)$ a $i_{20} = \sum_{k=1}^M (x_k - x_c)^2$, kde M je počet bodů uvnitř hranice a (x_c, y_c) jsou souřadnice centroidu regionu. [16]

	Kulatost	Konvexnost	Plnost	Kompaktnost
	0,587	0,351	0,731	0,766
	0,584	0,483	0,782	0,764
	0,447	0,349	0,592	0,668
	0,589	0,497	0,714	0,768

Obr. 3.4: Srovnání změny parametrů kulatost, konvexnost, plnost a kompaktnost v závislosti na tvaru objektu. Převzato z [20].

Projekce

Projekce je rozdělena na horizontální a vertikální. Horizontální je definována jako počet pixelů v jednotlivých řádcích obrazu a vertikální jako počet v jednotlivých sloupcích. Tento parametr může pomoci pro určování rozměrů ve směrech os x a y . [23]

Eulerovo číslo

Některé objekty v obraze se mohou skládat z více než jednoho souvislého regionu a uvnitř souvislého regionu se může vyskytovat oblast, která do něj nepatří - „díra“. Eulerovo číslo poté vyjadřuje rozdíl mezi počtem regionů S , ze kterých se objekt skládá a děr uvnitř jednotlivých oblastí N a je vypočteno podle rovnice 3.17

$$\vartheta = S - N. \quad (3.17)$$

Může nabývat i záporných hodnot. [23]

3.7 Topologie objektu

Účinně může být popsán tvar objektu pomocí jeho topologie. „Skeletonizace je transformace části digitálního obrazu do podmnožiny originální části.“ [13] Ta nese velmi mnoho informací o topologické struktuře objektu, což může být velice vhodné pro jeho popis a rozpoznání. Jedná se vlastně o reprezentaci objektu soustavou úseček, které nejlépe vyjadřují tvar objektu. Je-li vytvořena kostra, je možné původní tvar i přibližně zrekonstruovat. „Kostra je definována jako množina středů maximálních kružnic vepsaných segmentu“ [13] a skládá se z několika částí. Pixely pouze s jedním sousedním bodem jsou označovány jako koncové body, pixely s více než dvěma sousedními body se nazývají větvící body. Většina bodů má dva sousední, to jsou body, ležící na spojnici předchozích typů.

K popisu topologie může sloužit i pouhé spočítání větví, koncových bodů a smyček v topologii. Vyjádření objektů pomocí kostry využívá mnoho programů převádějících obrazy textu do textových souborů.

Pro digitální obrazy jsou na kostry kladeny následující čtyři požadavky, které musí být splněny [14] :

1. Kostra se musí nacházet uvnitř objektu
2. Kostra musí mít stejnou spojitost jako originální objekt a nesmí obsahovat žádné elementy na pozadí
3. Topologie musí zůstat nezměněná
4. Kostra musí být co nejúžší: pro 2D objekty je požadovaná tloušťka jeden pixel

3.7.1 Metody skeletonizace

Existuje několik různých kategorií algoritmů pro získávání skeletu objektu. První z nich je založena na vzdálenostech jednotlivých bodů od hranice regionu. Kostra je zde množinou bodů regionu, jenž jsou středy kružnic s maximálním poloměrem nacházejících se uvnitř regionu. Druhá kategorie se skládá ze ztenčovacích algoritmů. Zde dochází ke zmenšování hranice dokud není dosaženo tloušťky jeden pixel. Další kategorie je definována pomocí Voroného diagramu. [13] [26]

Vzdálenostní algoritmus

Tento algoritmus vytváří skelet pomocí symetrických bodů. Bod je nazýván symetrický, má-li od dvou různých hraničních bodů stejnou vzdálenost, která je rovna

nejmenší vzdálenosti tohoto bodu od hranice segmentu. Pozadí obrazu bývá reprezentováno pomocí nulových pixelů, takže vzdálenost může být počítána od nejbližších bodů s nulovou hodnotou.

Tomuto principu odpovídá Euklidovská vzdálenostní mapa. Ta každému pixelu uvnitř objektu přiřadí hodnotu, která vyjadřuje vzdálenost od nejbližšího pixelu pozadí, a tak vytvoří obraz ve stupních šedi. Počítat ovšem vzdálenost pro každou kombinaci bodů z pozadí a popředí by bylo značně neefektivní. Některé algoritmy tedy pracují jen ve čtyřech nebo osmi směrech.

Rychlejší metoda než počítání vzdáleností vychází s následujícího postupu [20] :

1. Každému pixelu pozadí je přiřazena hodnota 0
2. Proměnná N se nastaví na nulovou hodnotu
3. Každý bod dotýkající se pixelu s hodnotou N (ve čtyř- nebo osmi-směrovém sousedství) je nastavena hodnota $N + 1$
4. Proměnná N je inkrementována a opakuje se krok 3, dokud nemají všechny pixely obrazu přiřazenu hodnotu

Časově efektivnější postup byl představen v [6]. Již po dvou průchodech dává stejné výsledky jako předchozí.

1. Každému pixelu pozadí je přiřazena nulová a každému bodu uvnitř objektu velmi vysoká hodnota (musí být větší než maximální šířka regionu)
2. Prochází se zleva doprava shora dolů a každému pixelu v segmentu je nastavena hodnota o jedna vyšší než nejnižší hodnota sousedícího bodu
3. Opakuje se bod 2 s průchodem zprava doleva zespodu nahoru

Při těchto postupech jsou následně hledána lokální maxima. Problémem ovšem je, že v případě osmisměrového sousedství je i diagonální vzdálenost vyhodnocena hodnotou jedna. Toto odstraňují pozdější algoritmy fungující na podobném principu.

Ztenčovací algoritmus

Algoritmus ztenčování funguje na principu opakovaného odlupování hranice vrstvu po vrstvě. Jedná se o časově náročný proces avšak výhodou je uchování spojitosti skeletu. Nevýhodou je závislost pravidel pro odstraňování hraničních bodů na typu obrazu.

Existuje několik verzí tohoto algoritmu, některé mohou být nalezeny v [13].

Voroného algoritmus

Kostra pomocí Voroného algoritmu se počítá s využitím množiny hraničních bodů regionu. Základní princip Voroného algoritmu spočívá v rozdělení prostoru tak, že se v každém segmentu vyskytují pouze ty body, pro které je patřičný řídicí bod nejbližší. V tomto případě jsou řídicími body pixely nacházející se na hranici objektu. Výsledkem je vznik mozaiky, kde je strana každého segmentu tvořena body, které mají stejnou vzdálenost ke dvěma řídicím bodům. Vrcholy těchto segmentů označují pixely, jenž mají stejnou vzdálenost do tří a více řídicích bodů.

Nejprve je nutné připravit vstupní objekt a to tak, že se sníží počet hraničních bodů, tedy provede se aproximace hranice objektu pomocí polygonu. Kostra není příliš citlivá na nerovnosti hranice, proto to na výsledek nemá takový vliv, naopak je odstraněn šum na okraji regionu a také výpočet je mnohem jednodušší. Poté jsou pomocí algoritmu vypočítány Voroného segmenty. Jsou ponechány pouze ty, které se nacházejí uvnitř regionu. Vrcholy Voroného segmentů tvoří koncové a větvící body skeletu objektu. Jejich pospojováním, a to pomocí příslušných stran Voroného segmentů, vznikne kostra. [2]

3.8 Momenty

Rozpoznávání objektů v obrazech znesnadňuje fakt, že obraz je projekce trojrozměrného světa do dvojrozměrného plátna. Je-li tedy objekt vyfocen z různých úhlů, je nejen jeho výsledný tvar jiný díky existenci v reálném 3D světě, ale i jedna jeho strana, pokud je brána v úvahu například krychle, podléhá při zobrazení deformaci. Podrobné informace o využití momentů pro popis objektů a jejich výpočty, ze kterých je zde čerpáno, jsou uvedeny v literatuře [9].

Tento druhý problém, deformaci, je možné odstranit právě využitím momentů. „Momenty jsou skalární veličiny používané k charakterizaci funkcí a zachycení jejich charakteristických rysů.“ [9] Z matematického hlediska je moment projekcí funkce do polynomické báze. Geometrický moment digitální obrazové funkce řádu $r = p + q$ je dán rovnicí 3.18. [23]

$$m_{pq} = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} i^p j^q f(i, j), \quad (3.18)$$

kde i, j vyjadřují souřadnice pixelů. Geometrický moment je závislý na změně velikosti, posunutí, rotaci a transformaci obrazové funkce v každém bodě. Předpokladem tohoto výpočtu je nulová hodnota všech bodů pozadí. Geometrický moment je možné využít v případě binárního obrazu například k určení velikosti, k tomu slouží moment nultého řádu M_{00} . K určení centroidu, popsáno v kapitole 3.3, je možné využít následující rovnice:

$$x_c = \frac{m_{10}}{m_{00}}, y_c = \frac{m_{01}}{m_{00}}, \quad (3.19)$$

kde x_c a y_c vyjadřují souřadnice centroidu. [23]

Nezávislost na posunutí

Je-li pozice každého pixelu vztažena k pozici centroidu, jedná se o centrální geometrický moment, pomocí nějž lze dosáhnout nezávislosti na translaci. Jeho výpočet je uveden v rovnici 3.20. [23][9]

$$\mu_{pq} = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} (i - x_c)^p (j - y_c)^q f(i, j) \quad (3.20)$$

Pomocí centrálních momentů druhého řádu je možné vypočítat orientaci objektu. Vzorec pro vypočtení je uveden v rovnici 3.21. [11]

$$\Theta = \frac{1}{2} \arctan \left(\frac{2\mu'_{11}}{\mu'_{20} - \mu'_{02}} \right) \quad (3.21)$$

Nezávislost na změně velikosti

Nezávislosti na změně velikosti je možné dosáhnout pomocí normalizace každého momentu. Jako normalizační faktor může být využit jakýkoliv moment, nejčastěji je ovšem použit moment nultého řádu. Normalizace je uvedena v rovnici 3.22. [9]

$$\nu_{pq} = \frac{\mu_{pq}}{\mu_{pq}^{\gamma}}, \quad (3.22)$$

kde

$$\gamma = \frac{p + q}{2} + 1. \quad (3.23)$$

Nezávislost na otočení

Je-li požadována nezávislost na otočení objektu, je vhodné použít komplexní moment – 3.24. V této rovnici jsou souřadnice bodů vyjádřeny pomocí proměnných x a y , zatímco j zde označuje komplexní jednotku. Dříve byly k tomuto účelu používány Huova metoda, které ovšem v několika případech selhávala. [9]

$$c_{pq} = \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} (x + jy)^p (x - jy)^q f(x, y) \quad (3.24)$$

Komplexní moment je lepší pro tyto účely vyjádřit v polárních souřadnicích, jak je uvedeno v rovnici 3.25. [9]

$$c_{pq} = \sum_{x=-\infty}^{\infty} \sum_{y=-\infty}^{\infty} r^{p+q+1} e^{j(p-q)\theta} f(r, \theta), \quad (3.25)$$

kde $r = \sqrt{x^2 + y^2}$, $\theta = \arctan(\frac{y}{x})$, $x = r \cos \theta$, $y = r \sin \theta$.

Je-li objekt otočen o úhel α , vede výpočet na rovnici 3.26. [9]

$$c'_{pq} = e^{j(p-q)\alpha} c_{pq} \quad (3.26)$$

Amplituda momentu je tedy nezměněna, jen fáze je posunuta o $(p - q)\alpha$.

Jsou-li uvažovány momenty řádu $r \geq 2$ je množina rotačně nezávislých momentů B následující: [9]

$$B = \left\{ \Phi(p, q) \equiv c_{pq} c_{q_0 p_0}^{p-q} \mid p \geq q \wedge p + q \leq r \right\}, \quad (3.27)$$

kde p_0 a q_0 jsou libovolné indexy takové, že pro ně platí $p_0 + q_0 \leq r$, $p_0 - q_0 = 1$ a $c_{p_0 q_0} \neq 0$ pro všechny přípustné obrazy, a Φ je rotačně nezávislý moment.

Pokud je r liché je velikost množiny B dána dle rovnice 3.28, v případě, že je r sudé, platí rovnice 3.29. [9]

$$B = \frac{1}{4} (r + 1) (r + 3) \quad (3.28)$$

$$B = \frac{1}{4} (r + 2)^2 \quad (3.29)$$

Množina rotačně nezávislých momentů B je složena z prvků:

$$\begin{aligned} \Phi(p_0, q_0) &= c_{p_0 q_0} c_{q_0 p_0}, \\ \Phi(0, 0) &= c_{00}, \\ \Phi(1, 0) &= c_{10} c_{q_0 p_0}^2, \\ \Phi(2, 0) &= c_{20} c_{q_0 p_0}, \\ \Phi(1, 1) &= c_{11}, \\ \Phi(3, 0) &= c_{30} c_{q_0 p_0}^3, \\ &\vdots \\ \Phi(r, 0) &= c_{r0} c_{q_0 p_0}^r, \\ \Phi(r - 1, 1) &= c_{r-1,1} c_{q_0 p_0}^{r-2}, \\ &\vdots \end{aligned} \quad (3.30)$$

Důkazy i podrobnější informace lze nalézt v [9].

4 NÁVRH ALGORITMŮ

Hlavním tématem této práce je seznámení se se současným stavem problematiky popisu objektů v obrazech a následný návrh a implementace algoritmů pro tento účel. Jedná se o velice rozsáhlou problematiku, která je zkoumána již několik desetiletí. Tato část zpracování obrazů následuje ihned po segmentaci, tzn. že výstupní data segmentace jsou vstupními daty popisu objektu.

Jednotlivé segmenty jsou reprezentovány pomocí binárních masek. Pro každý segment vzniká nová maska, která má hodnoty pixelů nacházejících se mimo patřičný objekt nulové. Hodnotám uvnitř objektu budou přiřazeny hodnoty maximální, v osmi-bitové reprezentaci tomu tedy odpovídá hodnota 255. Výše popsané masky, jejichž počet je stejný jako počet identifikovaných objektů v obraze, a původní obraz vstupující do digitálního zpracování jsou tedy vstupními daty, se kterými se zde bude dále pracovat.

Tato data je zde třeba popsat takovým způsobem, aby bylo možné dané objekty v obraze podle určitých algoritmů rozeznat a identifikovat.

4.1 Popisované vlastnosti

Ze všech zmíněných parametrů pro popis obecných objektů v obraze byly vybrány barva a tvar. Velikost, jak absolutní tak relativní, není pro obecné objekty vhodná, protože nevíme, v jaké vzdálenosti od ohniska kamery je ten který objekt umístěn. Umístění a orientace nejsou vhodné z důvodu, že pro obecné objekty neprozrazují nic o jejich vlastnostech. Barva, která je sice stejně jako předchozí závislá na podmínkách pořízení snímku, konkrétně na umístění a typu zdroje osvětlení, je mnohem vhodnější.

Umístění zdroje osvětlení neovlivňuje tolik výslednou barvu objektu, může měnit především její jasovou složku, ovšem barvonosné složky zůstávají přibližně stejné. Větší vliv na barvu má typ zdroje osvětlení, což zná každý ze své vlastní zkušenosti. Z fyzikálního hlediska má povrch tělesa tu barvu, kterou odráží nebo vyzařuje. V případě odrazu tedy závisí na složení spektra dopadajícího světla. [12] Pokud dopadající světlo nebude obsahovat tu složku, která odpovídá barvě povrchu, těleso ho nebude moci odrazit a jeho barva se tedy bude jevit pouze jako odstín šedi.

Na obecných obrazech je ovšem uvažován zdroj bílého světla (Sluneční záření), které obsahuje všechny složky viditelné spektra.

4.1.1 Barva

Pro popis barvy byl zvolen výpočet dominantní barvy.

Dominantní barva, popsána v kapitole 3.2.1, je obecně vyjádřena dle vztahu 3.1. Zde nebude uvažována prostorová homogenita, takže vyjádření bude zjednodušeno na vztah 4.1.

$$F = (c_i, p_i), \text{ kde } i = 1, 2, \dots, N. \quad (4.1)$$

Výpočet dominantní barvy bude prováděn v barevném prostoru *RGB*, ve kterém jsou reprezentována vstupní data, ovšem není náročné upravit algoritmus dle potřeby na jiný barevný prostor.

4.1.2 Tvar

Pro popis tvaru bylo jako vhodný parametr zvoleno měřítkové zakřivení (Curvature Scale Space - CSS). Jedná se o algoritmus vyjadřující tvar na základě hraničních bodů segmentu. Dle testů uvedených v [16] se jedná o velmi robustní techniku pro popis tvaru objektů, která je poměrně odolná proti šumu. Princip této metody je naznačen v kapitole 3.6.1 v části Měřítkové zakřivení.

Dle doporučení [16] je k těmto výsledkům ještě vypočtena také excentricita a kulatost uvedené v kapitole 3.6.2. Výpočet kulatosti je již proveden v použitém rozšíření RapidMineru, jenž je popsáno v další kapitole. Tento výpočet se ovšem liší od výpočtu dle standardu MPEG-7, je tedy také nutný vypočíst stejně jako algoritmus pro zjištění excentricity objektu.

5 IMPLEMENTACE ALGORITMŮ

5.1 Prostředí RapidMiner

Celý projekt, zabývající se zpracováním obrazů, je implementován v prostředí RapidMiner. Jedná se o přední světový open-source systém pro dolování dat. [19] Zde je navíc rozšířen o knihovnu *Image Processing Extension*, která podporuje dolování dat z obrazů. Proces zpracování dat je tvořen pomocí operátorů, z nichž každý provádí dílčí činnost. V jednotlivých případech jsou tedy potřeba různé operátory. Například proces popisu dominantní barvy bude tvořen třemi operátory, první načte barevný snímek, druhý provede segmentaci a reprezentaci jednotlivých segmentů pomocí masek a ve třetím bude vypočtena dominantní barva.

Každý operátor může obsahovat jeden, či více vstupních portů, které přijímají vstupní data, jenž mají být zpracována. Může také obsahovat jeden, či více výstupních portů, kterými předává výsledky své činnosti na vstup dalšího operátoru nebo na výstup celého zpracování. Většina operátorů také obsahuje volitelné rozšiřující položky, pomocí nichž je možné například volit, co konkrétně nebo s jakou přesností má být počítáno.

Všechny operátory musejí splňovat určitá pravidla. Musí: [4]

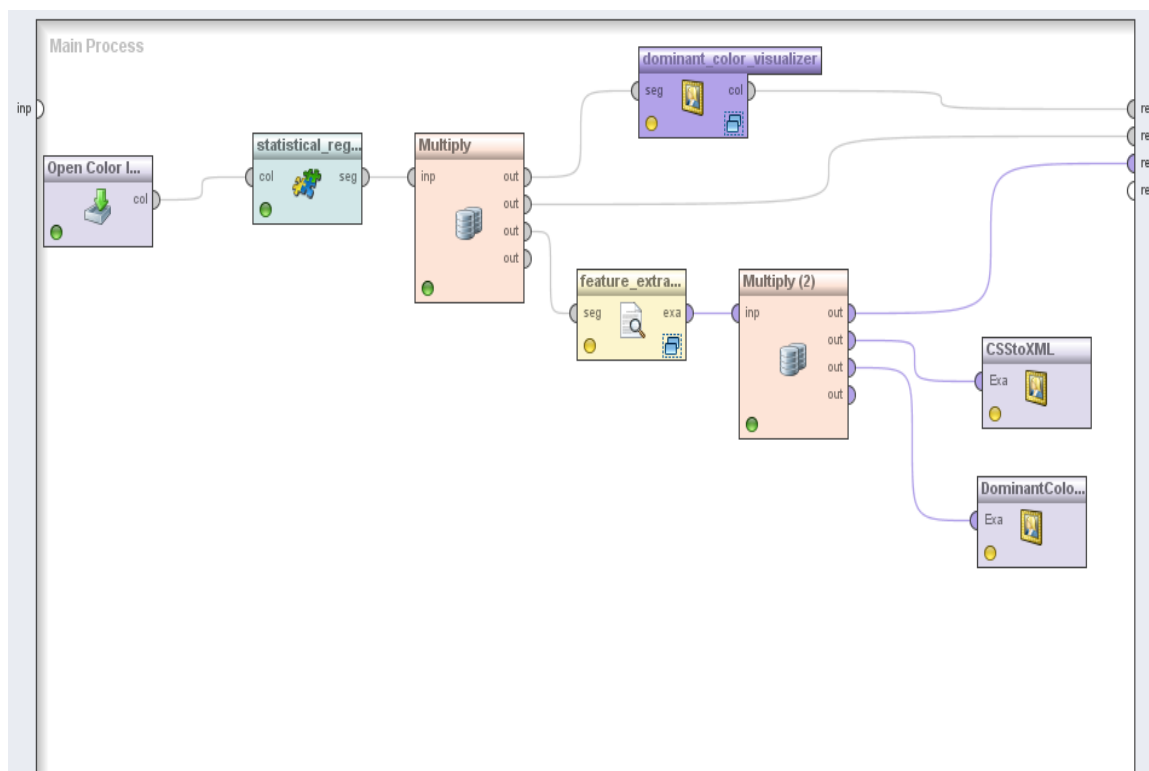
- být potomkem třídy *Operator*,
- obsahovat definici vstupních/výstupních portů, pokud jsou třeba
- v nich být obsažena metoda *doWork*
- zde být definovány uživatelské parametry, jsou-li třeba.

[4]

Metoda *doWork* obsahuje příkazy pro provádění dané činnosti, tedy výpočty a podobně. Pomocí uživatelských parametrů se například v operátoru *Roi Statistic* vybírají vlastnosti, které mají být pro jednotlivé objekty počítány. [4]

Ukázkové zapojení operátorů vytvořených, vylepšených nebo použitých v této práci je zobrazeno na obrázku 5.1. Operátor *Open Color Image* slouží k načtení zdrojového obrázku. Operátor *Multiply* pouze distribuuje data na více portů a zde i na celkový výstup, aby mohl být zdrojový obraz zobrazen. Operátor *Statistical Region Merging* provádí segmentaci, operátor *Feature Extractor* provádí označení jednotlivých *ROI* (Region Of Interest) a uvnitř něj je vnořen operátor pro výpočet vlastností jednotlivých objektů *Roi Statistic*. Operátor *Dominant Color Visualizer* provádí výpočet a vizualizaci jedné dominantní barvy pro každý objekt. Zbývající dva operátory zapisují hodnoty měřítkového zakřivení a dominantní barvy

do XML souborů. Porty *res*, na pravé straně obrázku, označují celkové výstupy procesu.



Obr. 5.1: Ukázkové zapojení operátorů

Výsledné výpočty mohou být pomocí speciálních operátorů vytvořených k tomuto účelu zapsány do XML souboru.

5.2 Dominantní barva

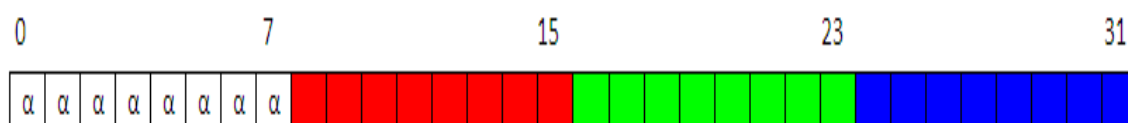
Při výpočtu dominantní barvy je zde především vycházeno z [16]. Dominantní barva je reprezentována vztahem 4.1, je tedy vyjádřena 3D vektorem v barevném prostoru *RGB* a procentuálním zastoupením v objektu. Vektor barvy je vypočten jako centroid klastru barev, tj. dané skupiny bodů v barevném prostoru.

Vstupem operátoru pro výpočet tohoto parametru je kromě segmentovaného a originálního obrazu také jedna uživatelem volitelná hodnota. Jedná se o počet dominantních barev, jež budou vypočítány, maximálně může mít tento parametr hodnotu 8, což bylo dle [16] stanoveno jako dostatečné pro popsání objektu.

Volitelná položka slouží jako podmínka pro výpočet. Jakmile je splněna, tedy počet barev je roven zvolenému počtu, je proces ukončen a výsledky jsou předány na výstup.

5.2.1 Postup výpočtu

Prvním úkolem při výpočtu je načtení všech barevných hodnot vyskytujících se v daném objektu. Jelikož je každý region obrazu reprezentován černobílou maskou, jedná se o body v místě, kde má tato maska bílou barvu. Po načtení je nutné oddělit jednotlivé hodnoty barev červené, zelené a modré, protože v paměti je barevný pixel vyjádřen pomocí dvaatřiceti bitového datového typu *int*. Dle [10] je zde každá barva reprezentována osmi bity, první osm je vyhrazeno pro tzv. α kanál, který je pro tyto výpočty nepodstatný, dalších osm bitů je vyhrazeno pro barvu červenou, třetí osmice pro barvu zelenou a zbývající bity vyjadřují modrou barvu, tak jak je naznačeno na obrázku 5.2.



Obr. 5.2: Reprezentace *RGB* v paměti

Jednotlivé barvy jsou načteny do dvojrozměrného pole, kde řádky reprezentují vyskytující se barvy. V prvních třech sloupcích jsou uloženy jejich souřadnice v *RGB* prostoru, zbylé sloupce jsou následně použity pro přiřazení barvy do klastru a uložení zkreslení daného bodu, tedy vzdálenost od centroidu. Poté je vytvořeno další dvojrozměrné pole, pro ukládání informací o dominantních barvách. Řádky opět reprezentují dané centroidy. První tři sloupce jsou určeny pro uložení souřadnic, dále je zde uchovávána hodnota zkreslení daného klastru, souřadnice nejvzdálenějšího bodu a procentuální zastoupení dané barvy v segmentu obrazu.

Jakmile jsou v poli zapsány všechny vyskytující se barvy, je přikročeno k samotnému výpočtu dominantní barvy. Nejprve je vypočten centroid všech zde se vyskytujících bodů v *RGB* prostoru. Toho je dosaženo vypočítáním průměrné hodnoty všech tří souřadnic. Tyto hodnoty jsou uloženy do pole centroidů. Poté je vypočteno také zkreslení a nejvíce zkreslený bod a opět jsou tyto hodnoty uloženy na příslušné místo pole centroidů. Všem bodům je zapsána stejná hodnota klastru a je pro vypočtena vzdálenost od zatím jediného centroidu.

V tuto chvíli je vypočtena jedna dominantní barva a její zkreslení. Všechny vyskytující se barvy v objektu spadají do stejné skupiny. Procentuální zastoupení této barvy je tedy stoprocentní. Pokud by byl vstupním parametrem stanoven výpočet pouze jediné barvy, je výpočet u konce.

Pokud předchozí podmínka splněna není, je vytvořen nový bod, kterým je nejvzdálenější bod klastru. Takto je vytvořen klastr nový. Následně je pro všechny

body původní skupiny spočtena vzdálenost od tohoto bodu. Pokud je nižší než vzdálenost od současného centroidu, je bod přiřazen do nově vzniklé skupiny, je-li vyšší, zůstává v původní. Pro původní i nový klastr je zopakován předchozí výpočet, tedy určen centroid, zkreslení a nejvzdálenější bod klastru a vypočteny vzdálenosti všech bodů skupiny od daného centroidu. Nakonec je nalezen klastr s největším zkreslením.

Dokud není splněna podmínka počtu barev je předchozí postup opakován vždy pro klastr s největším zkreslením.

Jakmile je výpočet ukončen, je přistoupeno k výpočtu procentuálního zastoupení jednotlivých dominantních barev v objektu a nakonec jsou obě výsledné hodnoty, tedy centroidy a procentuální zastoupení dominantních barev doručeny na výstup operátoru.

5.3 Měřítkové zakřivení

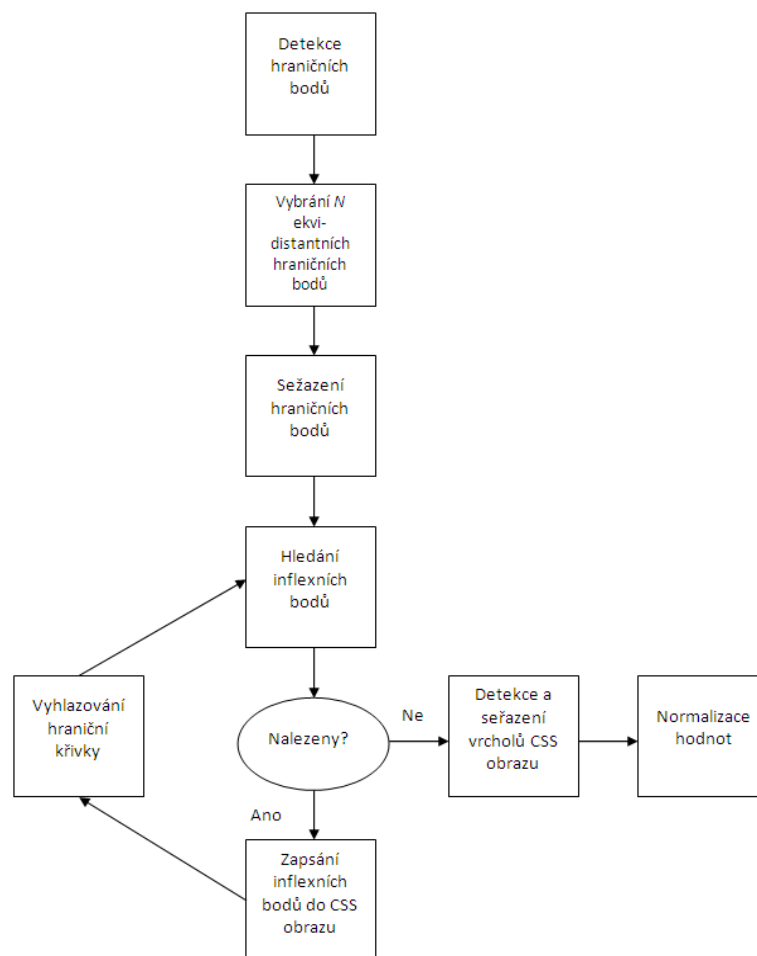
Jedná se o parametr popisující tvar jednotlivých segmentů obrazu. Je založen na rozdělení objektu na konvexní a konkávní oblasti v různých měřítkách.

5.3.1 Postup výpočtu

Prvním úkolem při získávání tohoto parametru je detekovat všechny hraniční body segmentu a seřadit je postupně ve směru hodinových ručiček. Použitý postup řazení částečně vychází z [3]. Následně je vybrán určitý počet ekvidistantních bodů a jsou nalezeny body inflexní. Poté je hraniční křivka opakovaně vyhlazována jednorozměrným gaussovským filtrem a v každém kroku jsou opět hledány inflexní body. Jakmile je hraniční křivka plně konvexní, tzn. nejsou zde žádné inflexní body, je proces ukončen. Body jsou seřazeny sestupně podle počtu potřebných filtrů a tyto hodnoty jsou normalizovány podle maximální hodnoty. Jsou nalezeny body splňující vstupní prahovou podmínku. Počet těchto bodů, maximální hodnota počtu filtrů a normalizované souřadnice těchto bodů v CSS obraze jsou doručeny na výstup. Průběh výpočtu je znázorněn na obrázku 5.3.

Detekce hraničních bodů

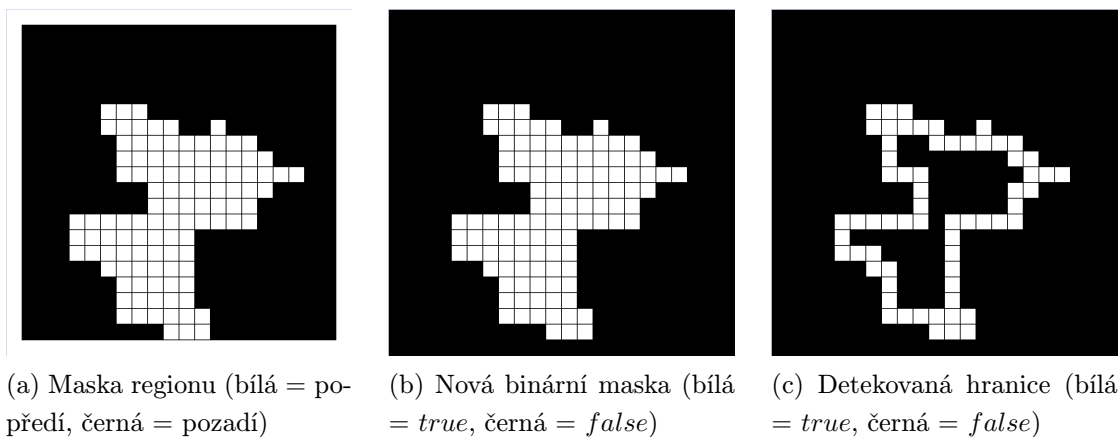
Nejprve jsou vytvořeny dva binární obrazy v každém směru o dva pixely větší než vstupní obraz. To z důvodu, aby nebylo třeba při průchodu kontrolovat, zda je aktuální pozice na okraji obrazu, či nikoliv, a tím pádem, ve kterých směrech není umožněno se pohybovat. Následně jsou body popředí masky přepsány do binárního obrazu masky tak, že jejich souřadnice v obou směrech jsou navýšeny o jedničku.



Obr. 5.3: Diagram výpočtu měřítkového zakřivení

Poté je procházen binární obraz masky a hledány hraniční body regionu, tedy takové body, jejichž alespoň jeden sousední pixel v osmi-okolí náleží pozadí. Pozadí je zde reprezentováno hodnotou *false*, popředí (maska regionu) hodnotou *true*. Bod binárního obrazu hranice se souřadnicemi detekovaných hraničních bodů je označen hodnotou *true*, tak jak je naznačeno na obrázku 5.4. Takto jsou do binárního obrazu zaneseny všechny hraniční body. Pro názornost jsou zde zobrazeny hranice mezi jednotlivými bílými pixely.

Jakmile jsou hraniční body takto zapsány, je přikročeno ke sledování okraje ve směru hodinových ručiček a zapisování horizontální i vertikálních souřadnic jednotlivých bodů do dvourozměrného pole hraničních bodů. Pro každý pixel jsou tedy uloženy dvě hodnoty - x a y .



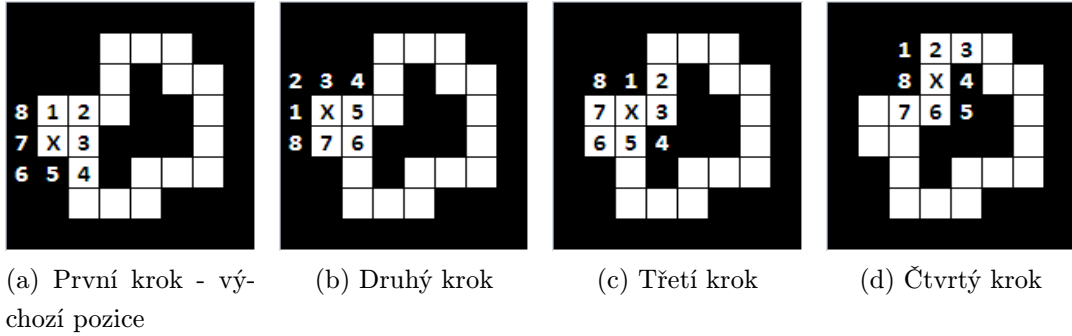
Obr. 5.4: Ukázka přepisu hraničních bodů do binárního obrazu a vytvoření hranice

Sledování okraje

Sledování okraje probíhá v binárním obraze detekované hranice a začíná u bodu s nejmenší x -ovou souřadnicí. Pokud je jich více, je jako výchozí bod zvolen ten z nich, který má největší y -ovou souřadnici. Z tohoto bodu je ve směru hodinových ručiček hledán sousední pixel. Jelikož je díky předchozím podmínkám jasné, že tento bod nemůže ležet vlevo od výchozího pixelu, začíná se prohledávat obraz ve směru kolmo nahoru. Jsou-li souřadnice výchozího bodu x, y , je nejprve vyhledáváno na pozici $x, y - 1$. Počátek souřadnic v obrazech se totiž nachází v levém horním rohu. Při vyhledávání je uvažováno sousedství typu osmi-okolí, proto, není-li zde nalezen hraniční bod, posune se vyhledávání kolem výchozího bodu o -45° , což je rovno jednomu pixelu ve směru hodinových ručiček. Nynější pozice se bude nacházet na $x + 1, y - 1$. Pokud ani zde není nalezen hraniční pixel, opět se posune pozice vyhledávání a to na $x + 1, y$. Takto je procházeno okolí pixelu, dokud není detekován hraniční bod. V tomto případě není uvažován případ, že by se jednalo o osamocení pixelu v obraze.

Jakmile dojde k nalezení sousedního hraničního pixelu, jsou jeho souřadnice zapísány do pole hraničních bodů na další pozici a je uložen směr, ve kterém byl detekován, tedy v jakém směru leží od výchozího pixelu. Poté je u něj proveden obdobný proces vyhledávání s tím rozdílem, že se nyní začíná na pozici otočené o 90° proti směru hodinových ručiček od uloženého směru. Pokud byl například sousední hraniční bod v předchozím kroku nalezen na pozici $x + 1, y$, začíná nové hledání na pozici $x', y' - 1$, kde x', y' jsou souřadnice aktuálního hraničního pixelu. Pro názornost je tento princip zobrazen na obrázku 5.5, kde písmenem X je vždy znázorněn aktuální hraniční pixel. Prošlé pixely nejsou nijak označeny, protože v určitých případech je možné jimi postupovat zpět. Čísly je vyjádřen postup při pro-

cházení sousedních pixelů. U třetího kroku je zřejmé, že některé pixely je také možné přeskočit a to z důvodu, že je uvažováno osmi-okolí.



Obr. 5.5: Ukázka přepisu hraničních bodů do binárního obrazu a vytvoření hranice

Tento proces je neustále opakován, dokud není detekovaný bod roven výchozímu pixelu při sledování hranice. V tomto případě cyklus končí a je nalezena celá hranice.

Z hraniční křivky je následně v závislosti na její délce vyčten určitý počet ekvivalentních bodů. To vede ke zjednodušení výpočtu, protože poklesne počet filtrování, avšak poměr počtu průchodů filtrem k odstranění jednotlivých inflexních bodů a normovaná pozice těchto bodů vzhledem k délce hranice zůstane stejná.

Detekce inflexních bodů

Aby mohli být detekovány inflexní body, je třeba v každém bodě, určeném v předchozím kroku, spočítat křivost. Výpočet vychází z rovnice 3.8. Pro detekci inflexních bodů je dle [16] nutné nalézt změnu znaménka křivosti sousedních bodů. Stačí tedy v každém bodě určit pouze znaménko křivosti, proto není třeba počítat jmenovatel této rovnice, který je vždy kladný, a výpočet v daném bodě je zjednodušen na rovnici 5.1, kde x', y' představují první a x'', y'' druhou derivaci souřadnice x , respektive y , podle pozice na hraniční křivce s . [16] V digitálním zpracování signálu je derivace definována jako rozdíl mezi současným a předchozím prvkem, podobně druhá derivace je rovna rozdílu prvních derivací v sousedních prvcích.

$$\kappa_z(s) = \text{sign}(x'(s)y''(s) - x''(s)y'(s)) \quad (5.1)$$

Jsou tedy vždy srovnány hodnoty znaménka křivosti sousedních prvků, jsou-li rozdílné, jedná se o inflexní bod. Takto nalezené body jsou následně uloženy do nového binárního CSS obrazu (tzv. *CSS image*), jehož horizontální souřadnice odpovídá pozici na hraniční křivce a vertikální souřadnice nese informaci o počtu použitých filtrů při detekování určitého inflexního bodu.

Protože může nastat případ, že křivost v daném bodě je nulová, je takovým bodů zapsáno znaménko předchozího pixelu.

Vyhlazování hranice

Proces vyhlazování může být prováděn různými způsoby a také různými filtry zvlášť pro horizontální a zvlášť pro vertikální souřadnice. Zde bylo zvoleno iterativní filtrování jednorozměrným gaussovským filtrem s jádrem přibližně $[0,274; 0,452; 0,274]$. Druhým způsobem je generování rozšiřujícího se gaussovského filtru a jeho aplikace stále na stejné hodnoty hraničních bodů. Filtrování je v obou případech prováděno do té doby, dokud nedojde k úplnému vyhlazení křivky. V každém kroku jsou vypočteny inflexní body a jsou zaznamenány do CSS obrazu.

Výstup

Posledním krokem výpočtu měřítkového zkreslení je nalezení lokálních maxim v obrazu CSS obrazu a jejich seřazení sestupně podle vertikální souřadnice, tedy potřebného počtu filtrů. To je dosaženo detekcí *true* pixelů se souřadnicemi x, y , pro které neexistuje *true* pixel na souřadnicích $x - 1, y$, $x - 1, y - 1$, $x, y - 1$ a $x + 1, y - 1$ a z důvodu možného vzniku nespojitosti v CSS obraze také na pozicích $x - 2, y - 1$ a $x - 2, y$.

Následně je nalezeno globální maximum v CSS obraze a hodnoty lokálních maxim jsou normalizovány vzhledem k této hodnotě. Poté jsou odfiltrovány inflexní body, které nesplňují prvotní podmínku normalizovaného počtu filtrů. Body, které ji splňují jsou doručeny na výstup spolu s jejich počtem a hodnotou globálního maxima, tedy amplitudě nejvyššího vrcholu v obraze.

Pro každý objekt je počet výstupních hodnot rozdílný, protože každý segment obrazu obsahuje jiný počet konkávních oblastí, čehož je také pro detekci objektů využíváno. Může ovšem nastat situace, že všechny objekty budou mít stejný počet výstupních vrcholů a to v případě, že bude zvolena prahová hodnota 100%. V tom případě by od sebe nebylo možné rozeznat ani velmi odlišné objekty, protože jak bylo popsáno v kapitole 3.6.1 v části věnované tomuto algoritmu, je zde využito nezávislosti na otočení a měřítku. Zde ovšem pomohou pro rozpoznání právě další doplňkové hodnoty vyjadřující tvar. Jsou jimi excentricita a kulatost, jejichž implementace je popsána v další části. I v tomto případě by ovšem mohlo dojít k nerozeznání méně odlišných objektů, proto je maximální hodnota 100% pouze teoreticky možnou, avšak určitě nevhodnou. Z testovacích důvodů je zde ale povolena.

5.4 Excentricita

Jedná se o další parametr charakterizující tvar, tentokrát ne pomocí hraničních bodů, nýbrž pomocí všech bodů regionu. Výpočet je proveden na základě rovnice 3.16. Nejprve je ovšem třeba nalézt centroid dle rovnice 3.2. Poté může teprve být

spočítána excentricita. Do výpočtu jsou zahrnuty všechny pixely, ležící uvnitř hranice segmentu, tedy ty, které jsou v masce segmentu vyjádřeny bílou barvou.

5.5 Kulatost

Tento parametr se dle standardu MPEG-7 počítá odlišně než v kapitole 3.6.2. Dle MPEG-7 je výpočet prováděn podle rovnice 5.2

$$k = \frac{o^2}{S}, \quad (5.2)$$

kde k značí kulatost, o obvod a S obsah daného objektu. [16] Protože výpočet obvodu a obsahu je již v použitém rozšíření RapisMineru proveden, je implementace algoritmu pro výpočet kulatosti triviální záležitostí.

5.6 Zápis do XML dle MPEG-7

Posledním úkolem této práce bylo zapsat výsledky předchozích výpočtů do XML souboru. Pro tuto činnost je třeba vytvořit nový operátor, který bude na vstupní port přijímat právě výstupní data operátorů počítajících měřítkové zakřivení, kulatost, excentricitu a dominantní barvu jednotlivých objektů.

Vytvořené operátory, sloužící k této činnosti, obsahují jeden vstupní port. Výstupní port se zde nevyskytuje žádný. Na vstupní port jsou doručena data z operátoru *Feature Extractor*, uvnitř nějž se nachází operátor *Roi Statistic*, který provádí výpočet různých charakteristik jednotlivých objektů. Data jsou doručena v tzv. *example set*, což je množina získaných výsledků pro jednotlivé objekty. Její struktura je následující. Pro každý ROI (oblast zájmu = segment obrazu) je v ní uloženo pole atributů, které byly vypočítány. V tomto případě je nutné, aby pole obsahovalo atributy náležející měřítkovému zakřivení nebo dominantní barvě. To je ihned zkontrolováno. Pokud není podmínka splněna, je ihned proces ukončen s vypsáním uživatelské chyby, která oznámí, který atribut přesně chybí.

Dále operátor obsahuje povinnou vstupní položku, kterou je specifikováno umístění vytvářeného XML souboru. Pokud zůstane položka prázdná, dojde ke stejnému výsledku, jako při chybějícím atributu, tentokrát s výpisem, že není specifikován výstupní soubor.

Následně je již přikročeno k vytváření XML souboru. K tomu museli být nainportovány speciální balíčky, díky nimž je tvorba těchto souborů umožněna. Těmi jsou *org.w3c.dom.Document* a *org.w3c.dom.Element*. Nejprve je nutné vytvořit novou instanci třídy *DocumentBuilderFactory*, poté nový objekt třídy *DocumentBuilder* (pro vytváření dokumentů) a konečně nový dokument.

Do vytvořeného dokumentu je zapsán kořenový prvek, jímž je element *MPEG7* s atributy definující schémata XML souborů. Tomuto prvku je přiřazen jeden potomek s názvem *DescriptionUnit*, který obsahuje atribut vyjadřující jeho typ a jehož hodnota je *DescriptorCollectionType*. Uvnitř tohoto elementu se již vyskytuje element *Image*, jehož atributem je název zdrojového souboru, např. *name = "africa.jpg"*. Ten obsahuje jednotlivé segmenty obrazu reprezentované prvky *roi* s atributem *id*, který upřesňuje daný region. Hodnota tohoto atributu je vyčtena z *example set*. Dále se operátory pro zápis měřítkového zakřivení a dominantní barvy liší a budou popsány v samostatných podkapitolách.

Následující podkapitoly popisují zápis jednotlivých vypočtených vlastností dle schématu [18]. K lepšímu pochopení přispěly ukázky XML souborů dle tohoto schématu uvedené v [21] a [1].

5.6.1 Zápis dominantní barvy

Zápis konkrétních parametrů dominantní barvy do XML souboru dle standardu MPEG-7 probíhá podle schématu uvedeného v [18]. Dle tohoto schématu je deklarace následující.

```
<complexType name="DominantColorType" final="#all">
  <complexContent>
    <extension base="mpeg7:VisualDType">
      <sequence>
        <element name="ColorSpace" type="mpeg7:ColorSpaceType"
          minOccurs="0"/>
        <element name="ColorQuantization"
          type="mpeg7:ColorQuantizationType" minOccurs="0"/>
        <element name="SpatialCoherency" type="mpeg7:unsigned5"/>
        <element name="Value" maxOccurs="8">
          <complexType>
            <sequence>
              <element name="Percentage" type="mpeg7:unsigned5"/>
              <element name="Index">
                <simpleType>
                  <restriction>
                    <simpleType>
                      <list itemType="mpeg7:unsigned12"/>
                    </simpleType>
                    <length value="3"/>
                  </restriction>
                </simpleType>
              </element>
            </sequence>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```

        </simpleType>
    </element>
    <element name="ColorVariance" minOccurs="0">
        <simpleType>
            <restriction>
                <simpleType>
                    <list itemType="mpeg7:unsigned1"/>
                </simpleType>
                <length value="3"/>
            </restriction>
        </simpleType>
    </element>
</sequence>
</complexType>
</element>
</sequence>
</extension>
</complexContent>
</complexType>

```

První komplexní typ umožňuje vytvořit element deskriptor s atributem *type* = "*DominantColorType*", který určuje typ daného obsahu a bude obsahovat vnořené elementy. Uvnitř něj již bude komplexní rozšiřitelný obsah se sekvencí daných hodnot pro určení dominantní barvy. Do této sekvence patří element určující barevný prostor *ColorSpace*, který je dle atributu *minOccurs* = "0" nepovinný. Dále se zde vyskytuje element vyjadřující kvantizaci barev, jenž obsahuje stejný parametr a je tedy také nepovinný. Zde, protože kvantizace použita nebyla, nebude uvažován.

Následující elementy jsou povinné. Jedná se o prvek nesoucí hodnotu celkové prostorové homogenity barev, jenž zde počítána nebyla a proto bude dle [16] obsahovat nulovou hodnotu. Dalším elementem je prvek *Value*, v němž jsou obsaženy již jednotlivé hodnoty pro konkrétní dominantní barvy. Tento prvek se může dle atributu *maxOccurs* = "8" vyskytovat maximálně osmkrát, což odpovídá uvedené dostatečnosti pro popis objektů či obrazů.

Uvnitř elementu *Value* je obsažena sekvence hodnot procentuálního zastoupení, vektoru barvy, který se skládá ze tří hodnot, což je určeno podmínkou *lengthvalue* = "3", a proměnlivosti barev, která se opět skládá ze tří hodnot. Pro každou barevnou složku může obsahovat hodnoty 0 nebo 1. Opět je dle atributu *minOccurs* = "0" vidět, že se jedná o parametr nepovinný a zde uváděn nebude.

Kvantizace

Aby mohli být hodnoty jednotlivých parametrů zapsány dle MPEG-7 standardu, je nutné provést jejich transformaci a kvantizaci. Ty vycházejí z [16]. Pro jejich lepší pochopení sloužily literatury [1],[15] a [21]. Hodnota vektorů barev zůstane stejná, jako je výsledek výpočtu, tedy vektor v RGB prostoru. Normalizované procentuální zastoupení jednotlivých dominantních barev je ovšem vyjádřeno pomocí pěti bitů. Maximální hodnota, pokud by bylo normalizované procentuální zastoupení rovno 1, je v případě pěti-bitového slova rovna 31.

Celý pravděpodobnostní prostor je tedy rozdělen na třicet jedna úseků a je provedena kvantizace. Výsledné hodnoty procentuálního zastoupení jednotlivých dominantních barev jsou v intervalu [1,31). Barvy, jejichž výsledné procentuální zastoupení po kvantizaci je rovno 0, do výsledku zahrnuty nejsou.

5.6.2 Zápis měřítkového zakřivení

Při zápisu měřítkového zakřivení se vychází ze stejného schématu a deklarace je uvedena následovně.

```
<complexType name="ContourShapeType" final="#all">
  <complexContent>
    <extension base="mpeg7:VisualDType">
      <sequence>
        <element name="GlobalCurvature"
          type="mpeg7:curvatureType"/>
        <element name="PrototypeCurvature"
          type="mpeg7:curvatureType" minOccurs="0"/>
        <element name="HighestPeakY" type="mpeg7:unsigned7"/>
        <element name="Peak" minOccurs="0" maxOccurs="62">
          <complexType>
            <attribute name="peakX" type="mpeg7:unsigned6"
              use="required"/>
            <attribute name="peakY" type="mpeg7:unsigned3"
              use="required"/>
          </complexType>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```


Zde se tedy opět vyskytuje komplexní typ, tentokrát atribut *name* = nese hodnotu "*ContourShapeType*". Uvnitř něj jsou obsaženy elementy nesoucí jednotlivé hodnoty. Globální zakřivení nese dle [16] informaci o kulatosti a excentricitě původního objektu. Prototypové zakřivení obsahuje stejné informace o vyhlazeném objektu. Tyto hodnoty jsou nepovinné a zde uvažovány nebyly. V elementu s názvem *HighestPeakY* (nejvyšší vrchol ve směru *y*) je uvedena absolutní hodnota nejvyššího vrcholu v CSS obraze, což odpovídá maximálnímu počtu použitých filtrů. Dále jsou jako atributy elementu s názvem *Peak* uvedeny normalizované souřadnice jednotlivých vrcholů, kterých dle atributu *maxOccurs* = "62" může být nejvýše šedesát dva.

Kvantizace

I v případě měřítkového zakřivení je nutné provést určité úpravy hodnot. Ty vycházejí stejně jako v předchozím případě z [16]. Nejprve je nutné změnit počáteční bod. Za počátek souřadnic je nyní považován bod s nejvyšší *y*-ovou souřadnicí v CSS obraze a *x*-ová pozice všem bodů jsou určovány relativně k němu. Stále je dodrženo udávání pozice ve směru hodinových ručiček, takže body, které byly nalevo od tohoto nejvyššího vrcholu, se objeví napravo a ve větší vzdálenosti než ostatní body. Tím se zaručí neměnnost měřítkového zakřivení vůči otočení.

Hodnoty pozic jsou poté vyjádřeny pomocí šesti bitů. Současný interval (0,1) je tedy rozdělen do 63 úseků a je provedena kvantizace hodnot.

Hodnoty *y*-ových souřadnic vrcholů CSS obrazů jsou vyjádřeny třemi bity. Jejich hodnota udává vždy relativní velikost vzhledem k předchozímu vrcholu. Vrcholy jsou totiž z výsledku seřazeny podle právě těchto hodnot. Výsledná kvantizace je tedy nelineární. Hodnota předchozího vrcholu je vždy rozdělena na sedm úseků a je provedena kvantizace. Je-li současný vrchol vzhledem k předchozímu tak malý, že po kvantizaci bude jeho hodnota nulová, je proces zapisování ukončen.

I hodnoty kulatosti a excentricity jsou kvantovány, každá je vyjádřena šesti bity. Pro hodnoty kulatosti je uvažován rozsah [12,110). Je-li hodnota vyšší, je oříznuta. V případě excentricity je rozsah hodnot [1,10). Překročí-li tuto hodnota, je proveden stejný krok, jako v případě kulatosti. Obě tyto hodnoty jsou kvantovány stejným způsobem jako předchozí, tedy interval je zde rozdělen na 63 úseků a je provedena kvantizace.

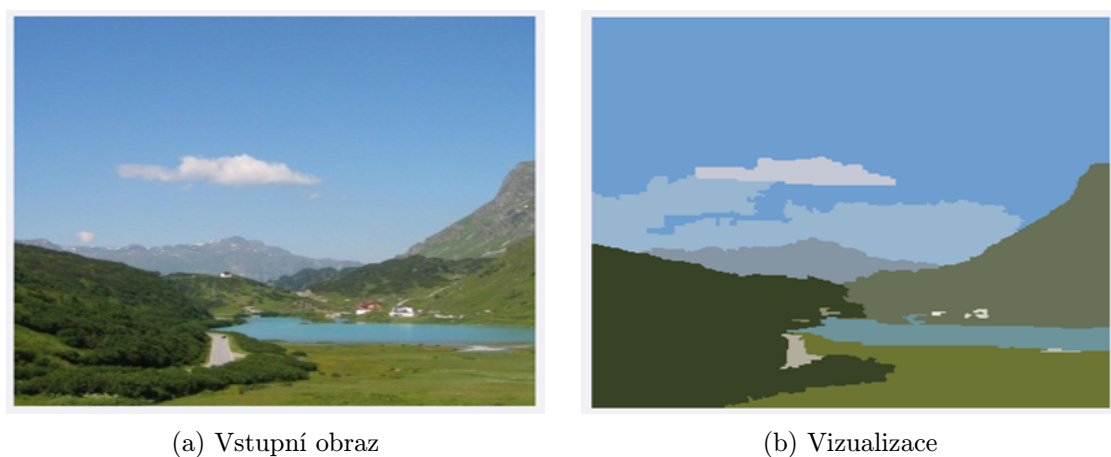
6 VÝSLEDKY

6.1 Dominantní barva

Výsledky výpočtu jsou uloženy do tzv. *example set*. Sem jsou ukládány vypočtené informace o nalezených segmentech obrazu. Pro vizualizaci výsledků byl vytvořen jednoduchý operátor, který na výstupu vytvoří obrázek, kde jsou masky jednotlivých segmentů vyplněny vypočtenou dominantní barvou. Je zde uvažován výpočet jediné dominantní barvy. Zkreslení v tomto případě není uvažováno. Výsledek je možné vidět na obrázcích 6.1 a 6.2.



Obr. 6.1: Vizualizace výpočtu jedné dominantní barvy



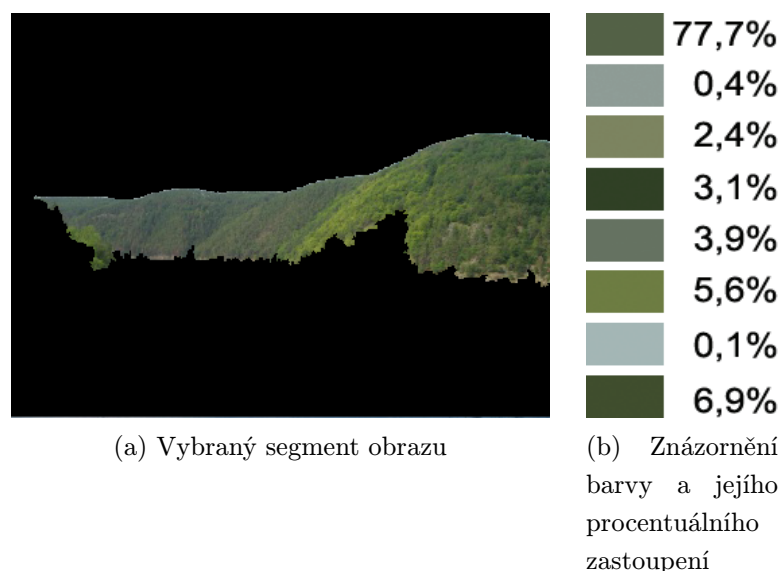
Obr. 6.2: Vizualizace výpočtu jedné dominantní barvy

Výsledný výpočet je ovlivněn přesností použité segmentační techniky, která předchází procesu popisu objektů. Pokud bude tato technika dostatečně přesná, tzn. že

do výsledného objektu budou zahrnuty pouze patřičné pixely, bude i výsledek přesnější. Toho je ovšem velmi složité dosáhnout, převážně u segmentů, které na sebe navazují plynule, jako například v obrázku 6.1 přechod z modrého nebe do mraků. Na tomto příkladě je patrné, že modrá barva nebe nalezeného vlevo dole je mírně tmavší než barva ostatního nebe. To je způsobeno právě zahrnutím bílých pixelů mraků do modrého nebe.

Tento problém je částečně odstraněn vypočtením více dominantních barev, přičemž jsou následně brány v úvahu především barvy s větším procentuálním zastoupením.

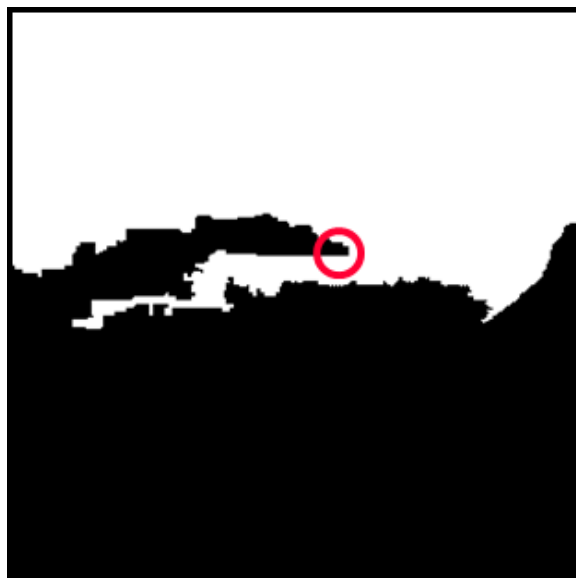
Výsledek výpočtu osmi dominantních barev pro jeden segment je znázorněn na obrázku 6.3, kde jsou znázorněny jak odstíny barev, tak jejich procentuální zastoupení a zkreslení.



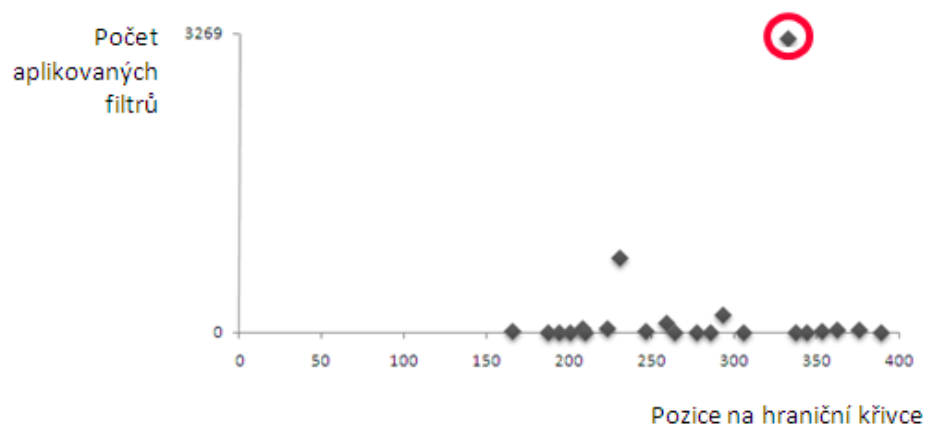
Obr. 6.3: Zobrazení výsledku výpočtu osmi dominantních barev pro jeden segment

6.2 Měřítkové zakřivení

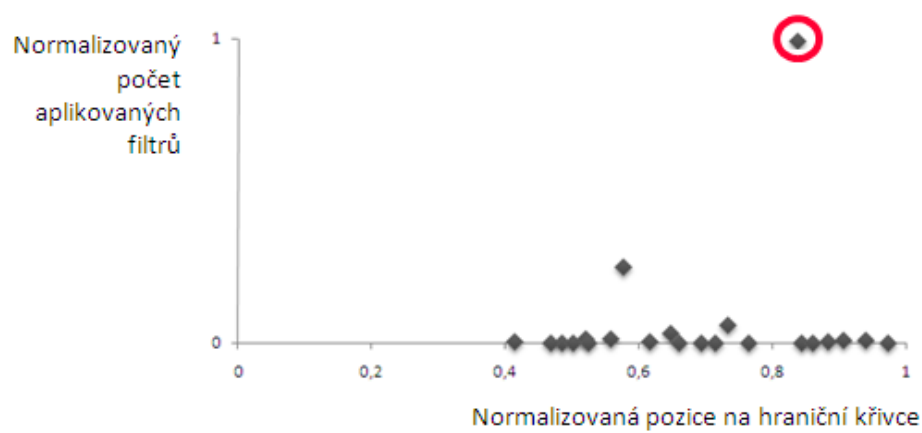
Obecným výsledkem výpočtu měřítkového zakřivení je CSS obraz. Pro uchování informací je ovšem důležitá především pozice jednotlivých vrcholů v tomto obraze. Příklad takovýchto bodů je zobrazen na obrázku 6.4. Červeným kolečkem je zde znázorněn bod, který leží v nejvíce konkávní oblasti a pro jeho odstranění bylo zapotřebí aplikovat největší počet filtrování. V obrázku 6.4b jsou zobrazeny absolutní hodnoty pozice a počtu filtrů, v obrázku 6.4c jsou tyto hodnoty normalizovány.



(a) Maska regionu



(b) Vrcholy CSS obrazu



(c) Vrcholy v normalizovaném CSS obraze

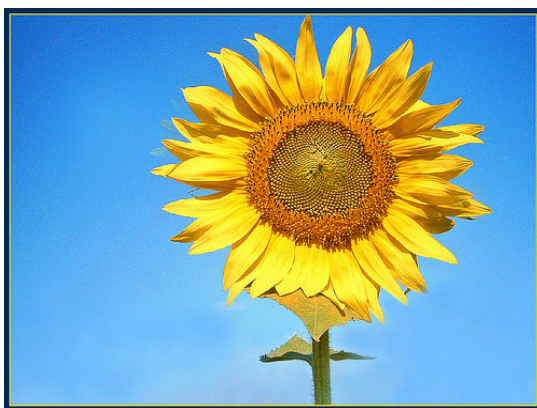
Obr. 6.4: Zobrazení vrcholů CSS obrazu

Normalizované hodnoty množství použitých filtrů jsou následně odfiltrovány v závislosti na vstupní prahové hodnotě. Zbylé normalizované hodnoty jsou spolu s maximální hodnotou aplikovaných filtrů a počtem vrcholů doručeny na výstup. Například pokud by byla prahová hodnota nastavena na 10%, což v normalizované hodnotě je rovno 0,01 a v případě absolutních hodnot tohoto příkladu rovno 32,69, byly by zde na výstup doručeny pouze dvě nejvyšší hodnoty.

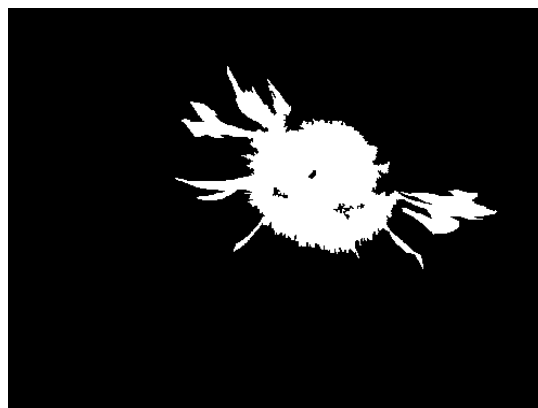
Výsledné hodnoty výpočtu měřítkového zakřivení jsou, stejně jako v případě dominantní barvy, ovlivněny přesností předchozí segmentační techniky. Jelikož je tento algoritmus poměrně odolný vůči šumu, není zkreslení výsledků tak významné. Jednotlivé pozice inflexním bodů na hraniční křivce jsou zkresleny pouze minimálně. Celkově je tato charakteristika odolná vůči šumu, ovšem při plynulých přechodech mezi objekty se mohou vytratit nebo objevit konkávní oblasti.

6.3 Zápis do XML

Zápis do XML souborů dle standardu XML byl úspěšný a jeho výsledky je možné vidět na obrázcích 6.6 a 6.7, jenž zobrazují částečné výstupní XML soubory pro obrázek 6.5a. Masku segmentu, jehož ID=60256, je zobrazena na obrázku 6.5b.



(a) Vstupní obraz



(b) Maska segmentu s ID=60256

Obr. 6.5: Vstupní obraz pro generaci XML souborů

```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <Mpeg7 xmlns="http://www.mpeg7.org/2001/MPEG-7_Schema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
- <DescriptionUnit xsi:type="DescriptorCollectionType">
- <Image name="213453205_2c1ee6193b.jpg">
- <Roi id="33358">
- <Descriptor xsi:type="ContourShapeType">
  <GlobalCurvature>63 9</GlobalCurvature>
  <HighestPeakY>263</HighestPeakY>
  <Peak peakX="34" peakY="6" />
  <Peak peakX="54" peakY="1" />
</Descriptor>
</Roi>

...

- <Roi id="60256">
- <Descriptor xsi:type="ContourShapeType">
  <GlobalCurvature>63 7</GlobalCurvature>
  <HighestPeakY>115</HighestPeakY>
  <Peak peakX="39" peakY="5" />
  <Peak peakX="13" peakY="6" />
  <Peak peakX="55" peakY="3" />
  <Peak peakX="31" peakY="7" />
  <Peak peakX="47" peakY="7" />
  <Peak peakX="20" peakY="4" />
  <Peak peakX="25" peakY="5" />
  <Peak peakX="52" peakY="5" />
</Descriptor>
</Roi>

...

</Image>
</DescriptionUnit>
</Mpeg7>

```

Obr. 6.6: Ukázka výstupního XML souboru s deskriptorem měřítkového zakřivení

```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <Mpeg7 xmlns="http://www.mpeg7.org/2001/MPEG-7_Schema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
- <DescriptionUnit xsi:type="DescriptorCollectionType">
- <Image name="213453205_2c1ee6193b.jpg">
- <Roi id="33358">
- <Descriptor xsi:type="DominantColorType">
  <ColorSpace>RGB</ColorSpace>
  <SpatialCoherency>0</SpatialCoherency>
- <Value>
    <Percentage>30</Percentage>
    <Index>229 173 23</Index>
    <Percentage>1</Percentage>
    <Index>88 34 9</Index>
  </Value>
</Descriptor>
</Roi>

...

- <Roi id="60256">
- <Descriptor xsi:type="DominantColorType">
  <ColorSpace>RGB</ColorSpace>
  <SpatialCoherency>0</SpatialCoherency>
- <Value>
    <Percentage>27</Percentage>
    <Index>180 113 17</Index>
    <Percentage>1</Percentage>
    <Index>242 225 114</Index>
    <Percentage>3</Percentage>
    <Index>61 24 7</Index>
  </Value>
</Descriptor>
</Roi>

...

</Image>
</DescriptionUnit>
</Mpeg7>

```

Obr. 6.7: Ukázka výstupního XML souboru s deskriptorem dominantní barvy

7 ZÁVĚR

Tato práce se zabývala digitálním zpracováním obrazů, konkrétně popisem detekovaných objektů. V úvodu jsou shrnuty důvody takového zpracování a následně prostudovány základní metody segmentace obrazů, jakožto procesu předcházejícího popisu objektů. Konkrétně jsou zde zmíněny metody: prahování, detekce hran, regionální metody jako rozšiřování či dělení oblastí a segmentace rozvodím.

Druhá kapitola je věnována standardu MPEG-7. Je zde popsán stručný úvod a charakteristika tohoto standardu. Hlouběji je do tohoto standardu proniknuto v kapitole věnující se vytváření jeho konkrétních deskriptorů a jeho zápisu do XML souborů.

Třetí kapitola se již plně věnuje problematice popisu objektů. Jsou shrnuty důležité vlastnosti detekovaných segmentů, jež je nutné nějakým způsobem charakterizovat. Jsou jimi především barva, tvar objektu, dále také jeho umístění, velikost či orientace. V této kapitole jsou shrnuty standardní i moderní metody.

V kapitole čtvrté jsou zhodnoceny možnosti popsané v předchozí části a vybrány algoritmy vhodné pro popis obecných objektů. Vhodnými parametry pro popis takovýchto objektů jsou tvar a barva, na které je zde zaměřena pozornost. Jako vhodný algoritmus pro popis barvy se jeví výpočet barvy dominantní. Těch se může v jednom segmentu obrazu nalézat větší množství. Jako dostačující hodnota bylo zvoleno číslo osm. Výstupními hodnotami těchto výpočtů je počet dominantních barev, jejich vektor v RGB prostoru a procentuální zastoupení v objektu.

Pro popis tvaru existuje robustní technika nazvaná měřítkové zakřivení (Curvature Scale Space), která byla pro charakterizaci objektů vybrána. Výstupem zmíněného algoritmu jsou: normalizovaná pozice bodů na hraniční křivce vzhledem k její délce, které leží uprostřed konkávních oblastí této křivky, normalizované množství vyhlazovacích procesů nutných k odstranění těchto oblastí vzhledem k maximální hodnotě, tato maximální hodnota a počet takto nalezených bodů (konkávních oblastí).

Pátá kapitola se věnuje implementaci zmíněných algoritmů do prostředí Rapid Miner a vytvoření operátorů pro zápis výsledků do XML souborů v souladu s MPEG-7. Algoritmy byly úspěšně aplikovány. Při aplikaci měřítkového zakřivení se zprvu jevilo, že výsledné hodnoty neodpovídají teoretickým předpokladům, vycházejících z [16] a [17], a potřebný počet aplikovaných filtrů je příliš velký. To bylo dle všeho způsobeno volbou příliš velkého množství ekvidistantních bodů. Z důvodu, že se jedná o poměrně náročný výpočet, byl počet těchto bodů stanoven na maximální hodnotu 100. Dle požadavků lze ovšem tento počet zvýšit.

Dalším problémem bylo pochopení zápisu hodnot do XML dle standardu MPEG-7. Bez ukázek těchto souborů v literaturách, které obsahují jejich ukázky, např.

[1] nebo [21], by bylo vytvoření standardizovaného souboru velmi obtížné. Taktéž obráceně, při čtení pouze těchto XML souborů bez znalosti přesných standardů MPEG-7 pro konkrétní deskriptory, není možná správná identifikace jednotlivých parametrů. Jejich hodnoty totiž neodpovídají obecný vyjádřením těchto parametrů.

Výsledky aplikace algoritmů na testovací obrázky jsou uvedeny v kapitole páté. Dle nich jsou algoritmy aplikované správně, tedy i funkce jednotlivých operátorů je správná.

Budoucím rozšířením této práce by mohla být implementace algoritmů pro porovnávání vypočtených hodnot s testovací databází, čímž by byla umožněna detekce konkrétních objektů.

LITERATURA

- [1] Arslan, S. *AN XML BASED CONTENT-BASED IMAGE RETRIEVAL SYSTEM WITH MPEG-7 DESCRIPTORS* [online]. 2004 [cit. 22. 5. 2011]. Dostupné z URL: <<http://etd.lib.metu.edu.tr/upload/12605750/index.pdf>>.
- [2] Burger, A.; Grana, M. *Pruning algorithm for Voronoi skeletons*, Univ Basque Country, GIC, Fac Informat San Sebastian, Paseo Manuel Lardizabal 1, San Sebastian 20018, Spain, 2010
- [3] Burger, W.; Burge, M. J. *Digital image processing: an algorithmic introduction using Java*, Springer, 2008, 560 s., ISBN 978-1-84628-379-6.
- [4] Burget, R., Karásek, J., Smékal, Z., Uher V., Dostál, O. *RapidMiner Image Processing Extension: A Platform for Collaborative Research*, International Conference on TELECOMMUNICATIONS AND SIGNAL PROCESSING, Baden Austria 2010.
- [5] Číka, P. *Multimédia, studijní materiály*, VUT v Brně, 2011.
- [6] Danielsson, P.E. *Euclidean distance mapping. Computer Graphics Image Process.* 1980.
- [7] Dougherty, G. *Digital Image Processing for Medical Applications.* [s.l.] : Cambridge University Press, 2009. ISBN 978-0-521-86085-7.
- [8] Dražić, S.; Ralevic, N.; Žunic, J. *Shape elongation from optimal encasing rectangles.* Computers and Mathematics with Applications, 2010.
- [9] Flusser, J.; Suk, T.; Zitová, B. *Moments and Moment Invariants in Pattern Recognition*, Wiley & Sons Ltd., 2009, 312 s., ISBN 978-0-470-69987-4.
- [10] *GImage*, WWW-CS-Faculty & Staff [online]. [cit. 24. 5. 2011]. Dostupné z URL: <<http://www-cs-faculty.stanford.edu/~eroberts/jtjf/javadoc/student/acm/graphics/GImage.html>>.
- [11] Chaumette, F. *Image Moments: A General and Useful Set of Features for Visual Servoing*, 2004
- [12] Jančovič, A. *Vnímání barev, Diplomová práce*, Masarykova univerzita v Brně [online]. 2005 [cit. 22. 5. 2011]. Dostupné z URL: <<http://www.ped.muni.cz/wphy/publikace/Jancovic1.html>>.
- [13] Kletten, G. *Skeletons in Digital Image Processing.* Computer Science Department of The University of Auckland, 2002.

- [14] Lakshmi, J. K.; Punithavalli, M. *A survey on skeletons in digital image processing*. Department of Computer science, SNR SONS college, Coimbatore 2009.
- [15] *M-OntoMat-Annotizer, Linking ontologies and multimedia low-level features for multimedia analysis, reasoning and retrieval, Tutorial* [online]. 2004 [cit. 23. 5. 2011]. Dostupné z URL: <<http://www.acemedia.org/aceMedia/files/software/m-ontomat/m-ontomat-annotizer-v0.60-tutorial.pdf>>.
- [16] Manjunath, B. S., Salambier, P., Sikora, T. *Introduction to MPEG-7: Multimedia content description interface*, Chichester, John Wiley & Sons Ltd. 2002, ISBN 978-0-471-48678-7
- [17] Mokhtarian, F., Bober, M. *Curvature Scale Space Representation: Theory, Applications, and MPEG-7*, Kluwer Academic Publisher, Dordrecht, ISBN 1-4020-1233-0
- [18] *Multimedia Content Description Interface, MPEG-7 Version 1 Schema* [online]. 2001 [cit. 22. 5. 2011]. Dostupné z URL: <http://www.mpeg7.org/2001/MPEG-7_Schema/mpeg7-v1.xsd>.
- [19] *Rapid-I* [online]. 2011 [cit. 15. 5. 2011]. Dostupné z URL: <<http://rapid-i.com>>.
- [20] Russ, J. C. *The image processing handbook, fifth edition*. Boca Raton, USA : CRC Press, Taylor & Francis Group, 2007. 829 s. ISBN 978-0-8493-7254-4.
- [21] Stanchev, P.; Green, D. Jr.; Dimitrov, B. *MPEG-7: THE MULTIMEDIA CONTENT DESCRIPTION INTERFACE*, International Journal „Information Theories & Applications“ Vol.11 [online]. 2007 [cit. 22. 5. 2011]. Dostupné z URL: <<http://www.foibg.com/ijita/vol11/ijita11-2-p03.pdf>>.
- [22] Straka, S. *Segmentace obrazů, Diplomová práce* [online]. 2009 [cit. 25. 11. 2010]. Dostupné z URL: <http://is.muni.cz/th/72784/fi_m/dp.pdf>.
- [23] Šonka, M.; Hlaváč, V.; Boyle, R. *Image Processing, Analysis, and Machine Vision, 3rd Edition*. Toronto, Canada: Thomson Engineering, 2007, 829 s.
- [24] Španěl, M.; Beran, V. *Obrazové segmentační techniky, Přehled existujících metod* [online]. 2005, poslední aktualizace 19. 1. 2006 [cit. 25. 11. 2010]. Dostupné z URL: <<http://www.fit.vutbr.cz/~spanel/segmentace>>.

- [25] *The MPEG Home Page* [online]. 2004 [cit. 22. 5. 2011]. MPEG-7 Overview Dostupné z URL: <<http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm>>.
- [26] Tran, S.; Shih, L. *Efficient 3D binary image skeletonization*, Computational Systems Bioinformatics Conference, 2005. Workshops and Poster Abstracts. IEEE , vol., no., pp. 364-372, 8-11 Aug. 2005.
- [27] Verdaguer, S. L. *Color Based Image Classification and Description*, Spain, 2009. 58 s. Master thesis, Universitat Politècnica de Catalunya

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

B množina rotačně nezávislých moment;

C centroid

c kulatost

c_{pq} komplexní moment

d průměr

e excentricita

F dominantní barva

f plnost

h tětivové rozdělení

k kompaktnost

m_{pq} geometrický moment

o obvod

p průměr

r roztažení

S obsah

s šířka

t činitel tvaru

κ křivost

μ_{pq} centrální geometrický moment

ν normalizovaný moment

Φ rotačně nezávislá

Θ orientace

ϑ Eulerovo číslo

MPEG Moving Picture Experts Group

SEZNAM PŘÍLOH

A	Obsah přiloženého CD	62
A.1	Návod na spuštění nástroje RapidMiner	62

A OBSAH PŘILOŽENÉHO CD

- Rozšíření *ImageProcessingExtension* pro RapidMiner
 - `cz.vutbr.feec.imageprocessing.op.imageDescription` - tento balíček se všemi jeho podtřídami byl vytvořen v rámci této diplomové práce
 - `cz.vutbr.feec.imageprocessing.op.visualize` - v tomto balíčku byla v této diplomové práci vytvořena třída `DominantColorVisualizer` pro vizualizaci jedné dominantní barvy
 - `cz.vutbr.feec.imageprocessing.op.featureExtraction.roi` - v tomto balíčku, konkrétně v třídě nazvané `StatisticsRoiOperator`, byly v této diplomové práci vytvořeny algoritmy pro výpočet měřítkového zakřivení, excentricity, kulatosti dle standardu MPEG-7 a dominantní barvy.
 - Ostatní balíčky a třídy, z nichž některé vytvářeli ostatní pracovníci a studenti pracující na tomto projektu, musí být přiloženy z důvodu správné funkce všech výpočtů.
- dva vzorové XML soubory, jejichž části byly uvedeny v kapitole 6 - jeden pro měřítkové zakřivení, druhý pro dominantní barvu

A.1 Návod na spuštění nástroje RapidMiner

Podrobný návod na spuštění nástroje RapidMiner a na jeho rozšíření o výše zmíněné *ImageProcessingExtension* je uveden na webové stránce:

- <http://spl.utko.feec.vutbr.cz/en/how-to-develop-with-rapidminer>